

Introducing Autodesk Maya 2015



Introducing Autodesk[®] Maya[®] 2015

Introducing Autodesk[®] Maya[®] 2015

DARIUSH DERAKHSHANI





Acquisitions Editor: Mariann Barsolo Development Editor: Stephanie Barton

Technical Editor: KEITH REICHER
Production Editor: REBECCA ANDERSON
Copy Editor: KIM WIMPSETT

Editorial Manager: Pete Gaughan

Vice President and Executive Group Publisher: RICHARD SWADLEY

Associate Publisher: Chris Webb Book Designer: Caryl Gorska

Compositors: Kate Kaminski and Maureen Forys, Happenstance Type-O-Rama

Proofreader: Josh Chase, Word One New York

Indexer: TED LAUX

Project Coordinator, Cover: TODD KLEMME

Cover Designer: WILEY

Cover Image: Courtesy Dariush Derakhshani

Copyright © 2014 by John Wiley & Sons, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-1-118-86284-1 ISBN: 978-1-118-86260-5 (ebk.) ISBN: 978-1-118-86282-7 (ebk.)

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at http://www.wiley.com/go/permissions.

Limit of Liability/Disclaimer of Warranty: The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Web site may provide or recommendations it may make. Further, readers should be aware that Internet Web sites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services or to obtain technical support, please contact our Customer Care Department within the U.S. at (877) 762-2974, outside the U.S. at (317) 572-3993 or fax (317) 572-4002.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at http://booksupport.wiley.com. For more information about Wiley products, visit www.wiley.com.

Library of Congress Control Number: 2014930417

TRADEMARKS: Wiley and the Sybex logo are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. Autodesk and Maya are registered trademarks of Autodesk, Inc. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

10987654321

Dear Reader,

Thank you for choosing *Introducing Autodesk Maya 2015*. This book is part of a family of premium-quality Sybex books, all of which are written by outstanding authors who combine practical experience with a gift for teaching.

Sybex was founded in 1976. More than 30 years later, we're still committed to producing consistently exceptional books. With each of our titles, we're working hard to set a new standard for the industry. From the paper we print on, to the authors we work with, our goal is to bring you the best books available.

I hope you see all that reflected in these pages. I'd be very interested to hear your comments and get your feedback on how we're doing. Feel free to let me know what you think about this or any other Sybex book by sending me an email at contactus@wiley.com. If you think you've found a technical error in this book, please visit http://sybex.custhelp.com. Customer feedback is critical to our efforts at Sybex.

V- Will-

Best regards,

Chris Webb

Associate Publisher, Sybex

About the Author

Dariush Derakhshani is a VFX and CG supervisor and educator in Los Angeles, California. Dariush has been working in CG for more than 17 years and teaching classes in CG and effects production for close to 16. He is the best-selling author of a handful of books, including the popular *Introducing Maya* books. He is also co-creator of *Learning Autodesk Maya*: A *Video Introduction*, available from Wiley. You can find out information on this video series and more at www.koosh3d.com.

Dariush started using the Autodesk® AutoCAD® software in his architecture days and then migrated to using 3D programs when his firm's principal architects needed to show their clients design work on the computer. Starting with Alias PowerAnimator version 6, which he encountered when he enrolled in the University of Southern California Film School's animation program, and working for a short while in Autodesk® 3ds Max® before moving on to Maya® jobs, Dariush has been using Autodesk animation software for the past 18 years.

He received an MFA in film, video, and computer animation in 1997 from USC. Dariush also holds a BA in architecture and theater from Lehigh University in Pennsylvania and worked at a New Jersey architecture firm before moving to Los Angeles for film school. He has worked on feature films, music videos, and countless commercials as a 3D animator and VFX supervisor, garnering honors from the London International Advertising Awards, the ADDY Awards, the Telly Awards, and a nomination from the Visual Effects Society Awards. He is bald and has flat feet.

Acknowledgments

 $As\ this\ book$ goes into its 11th edition, I am thrilled that the *Introducing Maya* book is a favorite resource for students and teachers of the Autodesk* Maya* software. Education is the foundation for a happy life, and with that in mind, I'd like to thank the outstanding teachers from whom I have had the privilege to learn. You can remember what you've been taught—or, just as important, you can remember those who have taught you. I also want to thank my students, who have taught me as much as they have learned themselves. Juan Gutierrez, Victor J. Garza, Robert Jauregui, and Peter Gend deserve special thanks for helping me complete the models and images for this book. Thanks to my colleagues and friends at work for showing me everything I've learned and making it interesting to be in the effects business. Special thanks to HP for its support and keeping me on the cutting edge of workstations and displays. ■ Thanks to my editors at Sybex and the folks at Autodesk for their support and help and for making this process fun. Thanks to the book team for bringing it all together: Mariann Barsolo, Stephanie Barton, Rebecca Anderson, and Kim Wimpsett. My appreciation also goes tenfold to technical editor Keith Reicher. ■ Thank you to my mom and brothers for your strength, wisdom, and love throughout. And a special thank-you to my lovely wife, Randi, and our son, Max Henry, for putting up with the long nights at the keyboard; the grumpy, sleep-deprived mornings; and the blinking and buzzing of all my machines in our apartment. Family!

CONTENTS AT A GLANCE

Introduction xvii

Introduction to Computer Graphics and 3D 1 Chapter 1 Chapter 2 Jumping into Basic Animation Headfirst 9 Chapter 3 The Autodesk® Maya® 2015 Interface **39** Beginning Polygonal Modeling 79 Chapter 4 Chapter 5 Modeling with NURBS Surfaces and Deformers 143 Practical Experience! 175 Chapter 6 Chapter 7 Autodesk* Maya* Shading and Texturing 219 Introduction to Animation 295 Chapter 8 Chapter 9 More Animation! 331 Autodesk® Maya® Lighting 393 Chapter 10

Chapter 11 Autodesk® Maya® Rendering 433

Chapter 12 Autodesk® Maya® Dynamics and Effects 517

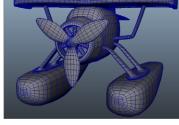
Index 563

Contents

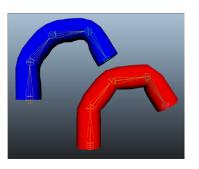
	Introduction	xvi
Chapter 1 ■ Int	roduction to Computer Graphics and 3D	1
	A Preview of the CG Process	2
	The CG Production Workflow	2
	Digital Images	4
	Summary	8
Chapter 2 ■ Jur	mping into Basic Animation Headfirst	9
	You Put the U in User Interface	10
	Project: The Solar System	14
	Creating a Project	15
	The Production Process: Creating and Animating the Objects	17
	Hierarchy and Maya Object Structure	28
	The Solar System, Resumed	30
	Outputting Your Work: Playblasting	37
	Summary	38
Chapter 3 ■ The	e Autodesk® Maya® 2015 Interface	39
	Navigating in Maya	40
	Exploring the Maya Layout	40
	Building a Decorative Box	50
	Mapping the Box's Reference Planes with Hypershade	53
	Organizing Workflow with the Layer Editor	57
	Modeling the Decorative Box	58
	Editing the Decorative Box Model Using the Shelf	63
	Continuing the Decorative Box Model	67
	Finishing the Decorative Box Model	71
	Summary	78



Chapter 4 ■ Be	ginning Polygonal Modeling	79
	Planning Your Model	80
	Polygon Basics	82
	Poly Editing Tools	84
	Putting the Tools to Use: Making a Cartoon Hand	93
	Creating Areas of Detail on a Poly Mesh	101
	Modeling a Catapult	115
	Summary	142
Chapter 5 ■ Mo	odeling with NURBS Surfaces and Deformers	143
	NURBS for Organic Curves	144
	Using NURBS Surfacing to Create Polygons	151
	Converting a NURBS Model to Polygons	153
	Using Artisan to Sculpt NURBS	154
	Creating a Pair of Glass Candle Holders	154
	Modeling with Simple Deformers	161
	The Lattice Deformer	166
	Animating Through a Lattice	170
	Summary	174
Chapter 6 ■ Pra	actical Experience!	175
	Evaluating the Toy Plane	176
	Building the Landing Pontoons	178
	Oh, What a Body! Modeling the Body of the Plane	194
	The Rear Stabilizers	198
	You Spin Me Right Round—The Engine and Propeller	201
	The Plane's Wings	206
	Assembling the Plane	215
	Summary	217



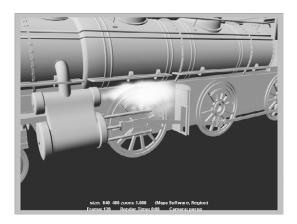
Chapter 7 ■ A	outodesk® Maya® Shading and Texturing	219
	Maya Shading	220
	Shader Types	220
	Shader Attributes	223
	Shading and Texturing the Toy Plane	227
	Textures and Surfaces	232
	Textures and UVs for the Red Wagon	240
	Photo-Real Mapping: The Decorative Box	274
	Toon Shading	291
	For Further Study	294
	Summary	294
Chapter 8 ■ II	ntroduction to Animation	295
	Keyframe Animation: Bouncing a Ball	296
	Throwing an Axe	307
	Replacing an Object	318
	Animating Flying Text	320
	Animating the Catapult	324
	Summary	329
Chapter 9 ■ N	Nore Animation!	331
	Skeletons and Kinematics	332
	Skeletons: The Hand	348
	Inverse Kinematics	362
	Basic Relationships: Constraints	366
	Basic Relationships: Set-Driven Keys	371
	Rigging the Locomotive	375
	Creating a Simple Character Rig	382
	For Further Study	391
	Summary	391



Chapter 10 ■ A	Autodesk® Maya® Lighting	393
	Basic Lighting Concepts	394
	Maya Lights	398
	Light Linking	405
	Adding Shadows	407
	Raytracing Soft Shadows	411
	mental ray Lighting	412
	mental ray Physical Sun and Sky	414
	Lighting Effects	418
	Assembling and Lighting a Scene	422
	Further Lighting Practice	430
	Tips for Using and Animating Lights	430
	Summary	432
Chapter 11 ■ Autodesk® Maya® Rendering		
	Rendering Setup	434
	Previewing Your Render: The Render View Window	443
	Setting the Default Renderer	445
	Reflections and Refractions	446
	Using Cameras	448
	Motion Blur	456
	Batch Rendering	456
	Rendering the Wine Bottle	458
	mental ray for Maya	461
	Render Layers	467
	Final Gather	478
	Ambient Occlusion	482
Y	HDRI	490
	Displacement Mapping the Decorative Box	493
	Rendering the Scene with mental ray	504
	Summary	516



Chapter 12 ■ Autodesk® Maya® Dynamics and Effects	
An Overview of Dynamics and Maya Nucleus	518
Rigid Bodies	518
Rigid Body Dynamics: Shoot the Catapult!	522
nParticle Dynamics	530
Emitting nParticles	530
Animating a Particle Effect: Locomotive Steam	538
Introduction to Paint Effects	547
Getting Started with nCloth	551
Customizing Maya	557
Summary	561
Where Do You Go from Here?	562
Index	563



Introduction

Welcome to Introducing Autodesk® Maya® 2015 and the world of computer-generated imagery (CGI). Whether you're new to 3D graphics or venturing into Autodesk's powerhouse animation software from another 3D application, you'll find this book a perfect primer. It introduces you to the Autodesk® Maya® software and shows how you can work with Maya to create your art, whether it's animated or static in design.

The first edition of this book was written out of the author's desire for solid, comprehensive, and yet open-ended teaching material about Maya for his classes. This book exposes you to all the facets of Maya by introducing and explaining its tools and functions to help you understand how Maya operates. In addition, you'll find hands-on examples and tutorials that give you firsthand experience with the toolsets. Working through these will help you develop skills as well as knowledge. These tutorials expose you to various ways of accomplishing tasks with this intricate and comprehensive artistic tool.

Finally, this book explains workflow. You'll learn not only how specific tasks are accomplished but why—that is, how they fit into the larger process of producing 3D animation. By doing that, these chapters should give you the confidence to venture deeper into the Maya feature set on your own or by using any of the other Maya learning tools and books as a guide.

It can be frustrating to learn a powerful tool such as Maya, so it's important to remember to pace yourself. The number-one complaint of readers of books like this is a sense that either the pace is too fast or the steps are too complicated or overwhelming. That's a tough nut to crack, to be sure, and no two readers are the same. But this book offers you the chance to run things at your own pace. The exercises and steps may seem challenging at times, but keep in mind that the more you try—even the more you fail at some attempts—the more you'll learn about how to operate Maya. Experience is the key to learning workflows in any software program, and with experience comes failure and aggravation. Nevertheless, try and try again, and you'll see that further attempts will be easier and more fruitful.

Above all, this book aims to inspire you to use Maya as a creative tool to achieve and explore your own artistic vision.

What You'll Learn from This Book

Introducing Autodesk Maya* 2015* will show you how Maya works and introduce you to every part of the toolset to give you a glimpse of the possibilities available with Maya.

You'll learn the basic concepts underlying animation and 3D and how to work with the Maya interface. You'll then learn the basic methods of modeling—creating objects and characters that appear to exist in three-dimensional space and that can be animated. You'll also explore shading and texturing—the techniques of applying surfaces to the objects you create—and you'll learn how to create lights and shadows in a scene. Animation is an enormously rich topic, but the practice and theory provided here will give you a solid footing. Next, you'll learn how to control the process of rendering, turning your images into files that can be viewed. Perhaps the most dazzling capability of Maya is its dynamics engine, software that allows you to make objects behave as if controlled by the real-world laws of physics.

After you've finished this book and its exercises, you'll have experience in almost everything Maya offers, giving you a solid foundation on which to base the rest of your Maya and CGI experience.

The goal of this book is to get you familiar enough with all the parts of Maya that you can work on your own and start a long, healthy education in a powerful and flexible tool. You will, however, learn the most from yourself.

Who Should Read This Book

Anyone who is curious about learning Maya or who is migrating from another 3D software package can learn something from this book. Even if you're highly experienced in another 3D package such as the open source Blender or Autodesk* 3ds Max* or Softimage*, you'll find this book helpful in showing you how Maya operates so you can migrate your existing skill set quickly and efficiently. By being exposed to everything Maya has to offer, you'll better understand how you can use its toolset to create or improve on your art and work.

If you already have cursory or even intermediate experience with Maya, culled from time spent learning at home, you can fill many holes with the information in this book as well as expand your experience. Self-education is a powerful tool, and the more you expose yourself to different sources, opinions, and methods, the better educated you'll be.

In addition, this book is invaluable for teachers in the CG field. This book was written to cater to those who want to pick up the fundamentals of Maya as well as those who want

to teach classes based on a solid body of course material. You won't find a better basis for a class when you combine this book with your own curriculum.

How to Use This Book

To begin reading this book, open it to some page, and read.

Introducing Autodesk® Maya® 2015 approaches the subject in a linear fashion that tracks how most animation productions are undertaken. But the book has numerous cross-references to make sure the chapters make sense no matter in which order you want to tackle them. You can open this book to any chapter and work through the tutorials and examples laid out for the Maya task being covered. Feel free to browse the chapters and jump into anything that strikes your fancy. However, if you're completely new to CG, you may want to take the chapters in order.

Although you can learn a lot just by reading the explanations and studying the illustrations, it's best to read this book while you're using Maya 2015 so that you can try the exercises for yourself as you read them. If you don't already have Maya, you can download an educational license if you are a qualifying student or faculty member at www.students. autodesk.com/ or a 30-day trial version of the software at www.autodesk.com/maya. This book refers to a companion web page (www.sybex.com/go/introducingmaya2015), containing all the example and support files you'll need for the exercises in the text as well as some videos to help you through some of the tutorials, which is a valuable educational aid. These videos will be updated over time, so make sure to check back in now and then for more material. You can use the example files to check the progress of your work, or you can use them as a starting point if you want to skip ahead within an exercise. The latter can save the more experienced reader tons of time. You'll also find it valuable to examine these files in depth to see how scenes are set up and how some of the concepts introduced in the book are implemented. Because Maya is a complex, professional software application, the exercises are both realistically ambitious and simple enough for new users to complete. Take them one step at a time and find your own pace, accepting aggravations and failures as part of the process. Take your time; you're not working on deadline—yet.

How This Book Is Organized

Chapter 1, "Introduction to Computer Graphics and 3D," introduces you to common computer graphics concepts to give you a basic overview of how CG happens and how Maya

relates to the overall process. In addition, it describes basic animation concepts to better build a foundation for further study.

Chapter 2, "Jumping into Basic Animation Headfirst," creates a simple animation to introduce you to the Maya interface and workflow and give you a taste of how things work right away. By animating the planets in our solar system, you'll learn basic concepts of creating and animating in Maya and how to use its powerful object structure.

Chapter 3, "The Autodesk* Maya* 2015 Interface," presents the entire Maya interface and shows you how it's used in production. Beginning with a roadmap of the screen, this chapter also explains how Maya defines and organizes objects in a scene while you are set to the task of building a decorative box model.

Chapter 4, "Beginning Polygonal Modeling," is an introduction to modeling concepts and workflows. It shows you how to start modeling using polygonal geometry to create various objects, from a cartoon hand to a catapult, using some of the new tools incorporated into Autodesk Maya 2015.

Chapter 5, "Modeling with NURBS Surfaces and Deformers," takes your lesson in modeling a step further. It shows you how to model with deformers and surfacing techniques, using NURBS to create a glass candle jar to later light and render in a scene. You'll also learn how to create NURBS surfaces directly into polygon meshes easily.

Chapter 6, "Practical Experience!," rounds out your modeling lessons with a comprehensive exercise showing you how to model a child's toy airplane using polygons. The chapter also exposes you to the powerful File Referencing workflow available in Maya.

Chapter 7, "Autodesk" Maya" Shading and Texturing," shows you how to assign textures and shaders to your models. Using a toy wagon model, you'll learn how to texture it to look like a real toy wagon as well as lay out its UVs for proper texture placement. Then, you'll create detailed photo-realistic textures based on photos for the decorative box and toy plane models. You'll also learn how to use toon shading to achieve a cartoon look for your renders.

Chapter 8, "Introduction to Animation," covers the basics of how to animate a bouncing ball using keyframes and then moves on to creating and coordinating more complex animation—throwing an axe and firing a catapult. You'll also learn how to import objects into an existing animation and transfer animation from one object to another, a common exercise in professional productions. In addition, you'll learn how to use the Graph Editor to edit and finesse your animation as well as animate objects along paths.

Chapter 9, "More Animation!" expands on Chapter 8 to show you how to use the Maya skeleton and kinematics system to create a simple walk cycle. This chapter also covers how to animate objects by using relationships between them. A thrilling exercise shows you how to rig a hand for easier animation and then a locomotive model for automated animation, some of the most productive uses of Maya.

Chapter 10, "Autodesk" Maya" Lighting," begins by showing you how to light a 3D scene as you learn how to light the toy plane and box that you modeled and textured earlier in the book. It also shows you how to use the tools to create and edit Maya lights for illumination, shadows, and special lighting effects. The mental ray for Maya Physical Sun and Sky feature is explored in this chapter as an introduction to some sophisticated techniques for mental ray lighting.

Chapter 11, "Autodesk" Maya" Rendering," explains how to create image files from your Maya scene and how to achieve the best look for your animation using proper cameras and rendering settings. You'll work with displacement maps to create details in a model. You'll also learn about the Maya renderer, the Vector renderer, and Final Gather using HDRI and image-based lighting through mental ray for Maya, as well as raytracing, motion blur, and depth of field. You'll have a chance to render the table lamp and decorative box to round out your skills.

Chapter 12, "Autodesk" Maya" Dynamics and Effects," introduces you to the powerful Maya dynamics animation system as well as nParticle technology. You'll animate pool balls colliding with one another using rigid body dynamics, and you'll fire the catapult. Using nParticle animation, you'll also create steam to add to your locomotive scene. You will then be exposed to nCloth to create a tablecloth and a flag. This chapter also shows you how to use Paint Effects to create animated flowers and grass within minutes.

Hardware and Software Considerations

Because computer hardware is a quickly moving target and Maya now runs on three distinct operating systems (Windows 7/8, Linux, and Mac OS X), specifying which hardware components will work with Maya is something of a challenge. Fortunately, Autodesk has a "qualified hardware" page on its website that describes the latest hardware to be qualified to work with Maya for each operating system, as well as whether you're running the 32-bit or 64-bit version. Go to the following site for the most up-to-date information on system requirements:

Although you can find specific hardware recommendations on these web pages, some general statements can be made about what constitutes a good platform on which to run Maya. First, be sure to get a fast processor; Maya eats through CPU cycles like crazy, so a fast processor is important. Second, you need lots of RAM (memory) to run Maya—at least 4 GB, but 8 GB or more is better to have, especially if you're working with large scene files and are on a 64-bit system. Third, if you expect to interact well with your Maya scenes, a powerful video card is a must; although Maya will mosey along with a poor graphics card, screen redraws will be slow with complex scenes, which can quickly become frustrating. You may want to consider a workstation graphics card for the best compatibility (rather than a consumer-grade gaming video card). Several companies make entry-level through top-performing workstation cards to fit any budget. A large hard disk is also important—most computers these days come with huge drives anyway.

Fortunately, computer hardware is so fast that even laptop computers can now run Maya well. Additionally, even hardware that is not officially supported by Autodesk can often run Maya—just remember that you won't be able to get technical support if your system doesn't meet the company's qualifications.

Writing this Book with the zBook 14

As you consider the hardware requirements for running Autodesk Maya 2015, it's interesting to note that a good portion of this book was written and accomplished on an ultrabook sized laptop workstation from HP, the zBook 14 (Figure FM.1). This little laptop has been a powerful ally in working on the scenes used throughout this book, as well as the actual writing of the book.

With a strong Intel i7 2.1Ghz CPU, 16 GB of memory and an HD resolution 14" LED display, I was able to write this book and work on the scene files for the exercises anywhere I went, even crushed in a coach class airplane seat. The ultrabook's AMD FirePro M4100 adeptly powered the graphics and even ran a second 24" display without a hiccup.

This is the first thin and light ultrabook form factor I've seen that is capable of work-station graphics, and I'm grateful to HP for allowing me the pleasure of using the zBook 14 while writing this book. It allowed me to have the stability and power I need without having to be chained to my desk, or toting around a much heavier 15" or 17" mobile workstation.



Figure F.1
The HP zBook 14
was a great little
machine to run
Maya while traveling around!

Free Autodesk Software for Students and Educators

The Autodesk Education Community is an online resource with more than five million members that enables educators and students to download—for free (see website for terms and conditions)—the same software used by professionals worldwide. You can also access additional tools and materials to help you design, visualize, and simulate ideas. Connect with other learners to stay current with the latest industry trends and get the most out of your designs. Get started today at www.autodesk.com/joinedu.

The Next Step

By the time you finish *Introducing Autodesk*° *Maya*° *2015*, you'll have some solid skills for using Maya. When you're ready to move on to another level, be sure to check out other Maya books from Sybex at www.sybex.com and additional resources at www.koosh3d.com.

You can contact the author through www.koosh3d.com and Facebook at www.facebook.com/IntroMaya. You may also go to the book's web page for the project files and video tutorials from the book at www.sybex.com/go/introducingmaya2015.



Introduction to Computer Graphics and 3D

This book will introduce you to the workings of 3D animation, also called *computer graphics* (CG), with one of the most popular programs on the market, the Autodesk® Maya® software. It will introduce you to many of the features and capabilities of Maya with the intent of energizing you to study further.

The best way to succeed at anything is to practice. Go through the exercises in this book (more than once if you care to), and also think of exercises and projects that can take you further in your learning process.

This is not to say you can't be a casual visitor to working in CG—far from it. Playing around and seeing what you can create in this medium is just flat-out fun. Understanding your own learning pace is important.

Throughout this book, you'll learn how to work with Maya tools and techniques at a pace you set for yourself. When you're learning how to work with Maya, the most important concept is discovering how you work as an artist.

Learning Outcomes: In this chapter, you will be able to

- Develop an appreciation for the CG process
- Discern between different types of digital image files
- Recognize key terms and principles in film and animation

A Preview of the CG Process

Try not to view this experience as learning a software package but as learning a way of working to an end. It's hard to relax when you're trying to cram so much information into your brain. You should try not to make this experience about how a software program works but about how you work with the software. Maya is only your tool; you're the boss.

CG and CGI are the abbreviations for *computer graphics* and *computer graphics imagery*, respectively, and are often used interchangeably.

The process of creating in CG requires that you either model or arrange prebuilt objects in a scene, give them color and light, and render them through a virtual camera to make an image. It's a lot like directing a live-action production but without any actor tantrums.

A large community on the Web provides free and for-pay models that you can use in your scenes. Sites such as www.turbosquid.com, www.cgtextures.com, doschdesign.com, and www.archive3d.net can cut out a lot of the time you might spend creating all the models or textures for a scene.

After you build your scene in 3D using models, lights, and a camera, the computer *renders* the scene, converting it to a 2D image. Through setup and rendering, CGI is born—and, with a little luck, a CG artist is also born.

The CG animation industry inherited a workflow from the film industry that consists of three broad stages: preproduction, production, and postproduction. In film, *preproduction* is the process in which the script and storyboards are written, costumes and sets are designed and built, actors are cast and rehearse, the crew is hired, and the equipment is rented and set up. In the *production* phase, scenes are taped or filmed in the most efficient order. *Postproduction* (often simply called *post*) describes everything that happens afterward: the scenes are edited into a story; a musical score, sound effects, and additional dialogue are added; and visual effects may also be added.

Preproduction for a CG animation means gathering reference materials, motion tests, layout drawings, model sketches, and such to make the actual CG production as straightforward as possible. Production begins when you start creating models of characters, sets, and props from the storyboards, model sheets, and concept art. Then these *assets* are animated, and finally, shots are then lit and ready for rendering. Postproduction for a CG project is where all of a CG film's elements are brought together and assembled into the final form through editing, adding sound, and so on.

The CG Production Workflow

Modeling almost always begins the CG process, which then can lead into texturing and then to animation (or animation and then texturing). Lighting should follow, with

rendering pulling up the rear as it must. (Of course, the process isn't completely linear; you'll often go back and forth adjusting models, lights, and textures along the way.)

Modeling, the topic of Chapters 4 through 6, is usually the first step in creating CG. Downloading or purchasing models from the Internet can often cut down the amount of time you spend on your project, if you don't prefer modeling or texturing.

Knowing how an object is used in a scene gives you its criteria for modeling. Creating a highly detailed model for a faraway shot will waste your time and expand rendering times needlessly.

Because your computer stores everything in the scene as vector math as opposed to pixels (called raster), the term geometry refers to all the surfaces and models in a scene.

When the models are complete, it's a good idea to begin *texturing* and *shading*, the process of applying colors and material textures to an object to make it renderable. In Figure 1.1, an elephant model is shown, with textures applied to its lower body.

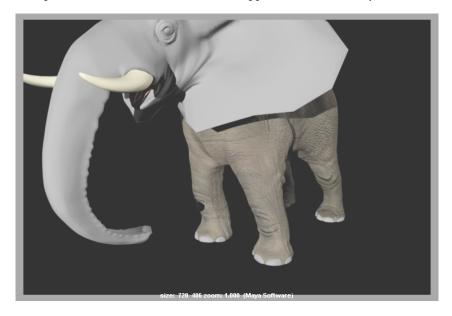


Figure 1.1 Texturing adds detail to an otherwise flat model.

Because the textures may look different after animating and lighting the scene, it's wise to leave the final adjustments for later. You'll learn more about texturing and shading in Chapter 7, "Autodesk Maya Shading and Texturing."

Next comes the process of animating. Animation shows change over time. All animation, from paper flipbooks to film to Maya, is based on the principle of persistence of vision that when we see a series of rapidly changing images (called *frames*), we perceive the changing of the image to be in continuous motion.

You know when something doesn't look right, and so will the people watching your animation. So, to animate something properly, you may need to do quite a lot of setup beyond just modeling and texturing. In character animation, for example, you'll need to create and attach an *armature*, or skeleton, to manipulate the character and to make it move like a puppet. I cover animation techniques in Maya in Chapter 8, "Introduction to Animation," and Chapter 9, "More Animation!"

Next there is lighting. *CG* is fundamentally all about light. Without light, we wouldn't see anything; simulating light is the one of the most influential steps in CG. During the lighting step, you set up virtual lights in your scene to illuminate your objects and action, as you'll learn in Chapter 10, "Autodesk Maya Lighting." Before long, you'll start modeling and texturing differently—that is, working with the final lighting of the scene in mind.

When you're done lighting, you move to rendering; your computer takes your scene and makes all the computations it needs to render your scene into images. Rendering time depends on how much geometry is used in the scene as well as on the number of lights, the size of your textures, and the quality and size of your output: the more efficient your scene, the shorter the rendering times. The more experience you gain, the more efficient your eye will become.

Digital Images

When you're finished with your animation, you'll probably want as many people as possible to see it (and like it!). Here is a primer on the types of digital images and how they are discerned.

As mentioned, each image file holds the color information in *channels*. All color images have red, green, and blue color channels. Some images have a fourth channel called the *alpha* channel (a.k.a. *matte* channel), which defines what portions of the image are transparent or opaque (solid). You can read more about alpha channels in Chapter 7.

First, an image file stores the color of each pixel as three values representing red, green, and blue. The image type depends on how much storage is allotted to each pixel (the *color depth*).

Grayscale The image is black and white with varying degrees of gray (typically 256 shades). These are good for use as texture maps such as bump and displacement maps.

8-Bit Image File (a.k.a. 24-Bit Color Display) Each color channel is given 8 bits for a range of 256 shades of each red, green, and blue channel, for a total of 16 million colors in the image.

16-Bit Image Used in television and film work with such file types as TIFF16, a 16-bit image file holds 16 bits of information for each color channel, resulting in an impressive number of color levels and ranges.

32-Bit Image 32-bit image files, such as the OpenEXR format, give you an incredible amount of range in each color channel.

File Formats

Several image file formats are available today. The main difference between file formats is how the image is stored. Some formats compress the file to reduce its size. However, as the degree of compression increases, the color quality of the image decreases.

The popular formats to render into from Maya are Tagged Image File Format (TIFF) and Targa. These file formats maintain a good 8-bit image file, include an alpha channel, and are either uncompressed or barely compressed (lossless compression) for high quality. To see an animation rendered in a file sequence of TIFFs or JPEGs, for example, you must play them back using a frame player, such as FCheck (which is included with Maya) or compile them into a movie file using a program such as Adobe After Effects.

Animations can also be output directly to movie files such as AVI or QuickTime. These usually large files are self-contained, and hold all the images necessary for the animation that they play back as frames. Movie files can also be compressed, but they suffer from quality loss the more they're compressed.

Maya can render directly to a movie format, saving you from having to render out a large sequence of files, though it's always best to render a sequence of files that can be compiled into a movie file later using a program such as Adobe After Effects, Adobe Premiere Pro, or QuickTime Pro.

Resolution, Aspect Ratio, and Frame Rate

Resolution denotes the size of an image by the number of horizontal and vertical pixels, usually expressed as #x# (for example, 640x480). The higher the resolution, the finer the image detail will be, and the larger the files. Full HD, for example, is defined as 1920x1080. Aspect ratio is the ratio of the screen's width to its height. Finally, the number of frames played back per second determines the *frame rate* of the animation. This is denoted as *frames per second* (fps). Most digital movie files are output at 24fps to mimic the frame rate of movies.

You should have your frame rate set properly before animating to match the frame rate you intend on using for playing the animation back. Playing back a 24fps animation at 30fps will show a slower-moving animation. Conversely, playing a 30fps animation at 24fps will create a faster-moving animation that will either skip some frames or end later than it should.

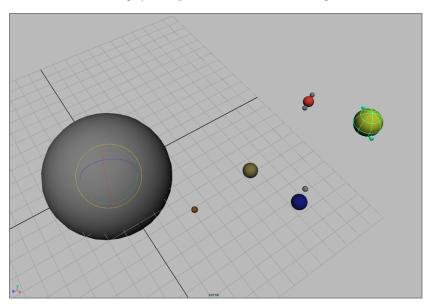
3D Coordinate Space, World Axis, and Local Axis

Space is defined in three axes—X, Y, and Z—representing width, height, and depth. The three axes form a numeric grid in which a particular point is defined by *coordinates* set forth as (#,#,#), corresponding to (X,Y,Z), respectively. At the zero point of these axes is the *origin*. This is at (0,0,0) and is the intersection of all three axes. The 3D space defined by these three axes is called the *World axis*, in which the XYZ axes are *fixed references*.

The axis in World Space is always fixed and is represented in Maya by the XYZ Axis icon in the lower-left corner of the Perspective window.

Because objects can be oriented in all sorts of directions within the World axis, it's necessary for each object to have its own width, height, and depth axis independent of the World axis. This is called the *Local axis*. The *Local axis* is the *XYZ*-coordinate space that is attached to every object in Maya. When that object rotates or moves, its Local axis rotates and moves with it. This is necessary to make animating an object easier as it moves and orients around in the World axis. Figure 1.2 shows an example of a Local axis in action, where we can see a large yellow planet and its moon rotating around the central sun.

Figure 1.2 The Sun at the origin, with Earth and other planets orbiting the World axis while rotating on their own axes



Basic Animation Concepts

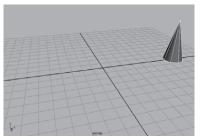
The following sections define the key terms you'll come across numerous times on your journey into animation and CG.

Frames, Keyframes, and In-Betweens

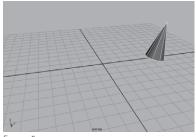
Each drawing of an animation—or, in the case of CG, a single rendered image—is called a frame. The term frame also refers to a unit of time in animation whose exact chronological length depends on how fast the animation will eventually play back (frame rate).

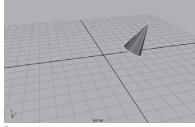
Keyframes are frames in which the animator creates a pose or other such state of being for an object or character. Animation is created when an object travels or changes from one keyframe to another. In CG, a keyframe can be set on almost any aspect of an object—its color, position, size, and so on. Maya then interpolates the *in-between* frames between the keyframes set by the animator. Figure 1.3 illustrates a keyframe sequence in Maya.

In the first frame of this sequence, a keyframe is set on the position, rotation, and scale of the cone. On frame 30, the same properties are again keyframed. Maya calculates all the movement in between.

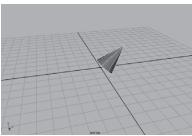


Keyframe at frame 1

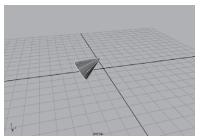




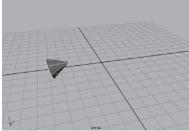
Frame 5



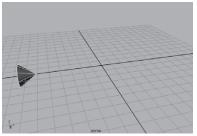
Frame 10



Frame 15



Frame 20



Frame 25

Keyframe at frame 30

Weight

Weight is an implied, if not critical, concept in design and animation. How you show an object in motion greatly affects its weight and therefore its believability. Weight in animation is a perception of mass. An object's movement, how it reacts in motion, and how it reacts to other objects together convey the feeling of weight. Otherwise, the animation will look bogus—or, as they say, "cartoonish."

Weight can be created with a variety of techniques developed by traditional animators over the years. Each technique distorts the shape of the object or character in some way to make it look as if it's moving. The following are a few animation principles to keep in mind:

Squash and Stretch This technique makes a character, for example, respond to gravity, movement, and inertia by squashing it down and stretching it up when it moves. For example, a cartoon character will squeeze down when it's about to jump up, stretch out a bit while it's flying in the air, and squash back down when it lands.

Ease-In and Ease-Out Objects never really stop suddenly; everything comes to rest in its own time, slowing before coming to a complete stop in most cases. This is referred to as ease-out. Objects don't immediately start moving either; they accelerate a bit before reaching full speed; this is referred to as ease-in. The bouncing-ball tutorial in Chapter 8 illustrates ease-in and ease-out.

Follow-Through and Anticipation Sometimes you have to exaggerate the weight of an object in animation, especially in cartoons. You can exaggerate a character's weight, for instance, by using well-designed follow-through and anticipation.

You should create a bit of movement in your character or object *before* it moves. *Anticipation* is a technique in which a character or object winds up before it moves, like a spring that coils inward before it bounces.

Likewise, objects ending an action typically have a follow-through. Think about the movement of gymnasts. When they land, they need to bend a bit at the knees and waist to stabilize their landing. In the same way, a cape on a jumping character will continue to move even after the character lands.

The axe tutorial in Chapter 8 will give you a chance to implement these two concepts.

FOR FURTHER REFERENCE

Some of the exercises in this book also have videos created for your use and may be found at: www.sybex.com/go/introducingmaya2015. For more video tutorials and reference on Maya, you can visit www.koosh3d.com.

Summary

In this chapter, you learned the basic process of working in CG, called a workflow. In addition, you were introduced to some fundamentals of digital images and animation principles.

Now that you have a foundation in CG and 3D terminology and core concepts, you're ready to tackle the software. Maya is a capable, intricate program. The more you understand how you work artistically, the better use you'll make of this exceptional tool. Have fun!



Jumping into Basic Animation Headfirst

In this chapter, you'll start using the Autodesk® Maya® software and get your groove on. This will be a quick primer on the Maya interface so you experience tasks right away. The next chapter will show you more details and provide additional explanations and a reference of how the entire Maya interface functions as you work on another project.

In this chapter you will follow an exercise for creating a solar system and learn the mechanics of animating orbits. With this exercise, you'll create simple objects, set keyframes, and stack your animation to get planets and moons to orbit each other and the Sun. This will expose you to object creation, simple modeling, object components, pivot-point placement, grouping and hierarchies, basic keyframing, and timing.

Learning Outcomes: In this chapter, you will be able to

- Gain a working understanding of the user interface and how to navigate in 3D space
- Learn project structure in Maya and how to create projects in Maya
- Create, name, and manipulate simple objects with the Move, Rotate, and Scale tools
- Make and apply simple shaders to scene geometry through the Hypershade
- Add keyframes to objects to create animation
- Adjust pivot points
- Create and edit hierarchies by using groups
- Output your animation through playblasting

You Put the *U* in User Interface

Fire up your computer, and let's get going. This section will introduce you to getting around the Maya user interface (UI).

The overall goal of this chapter is to expose you to Maya UI basics as well as important scene creation and editing tools. You'll find more details on the interface in Chapter 3, "The Autodesk Maya 2015 Interface."

KEYBOARD AND SYMBOL CONVENTIONS USED IN THIS BOOK

The following terms are used throughout this book:

Click and LMB+Click These refer to a mouse click with the primary (left) mouse button.

RMB+Click This refers to a mouse click with the right mouse button.

MMB+Click This refers to a mouse click with the middle mouse button.

Shift+Click This indicates you should hold down the Shift key as you click with the primary (left) mouse button.

Shift+Select This indicates you should hold down the Shift key as you select the next object for multiple selections.

The ☐ Symbol This, next to a menu command, indicates you should click the box (☐) next to the menu command to open the options for that command.

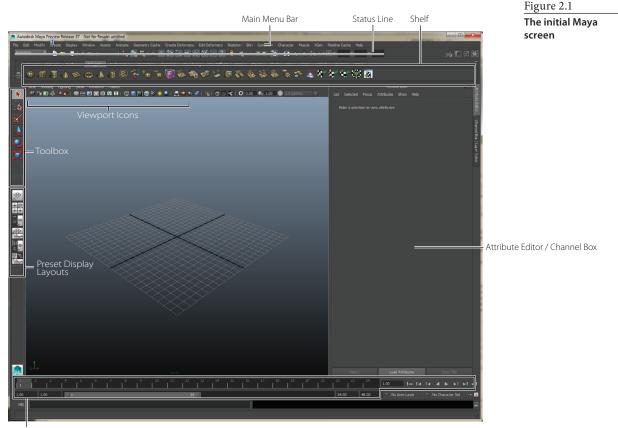
A Quick Screen Roadmap

Let's get to the basics of how Maya is laid out (see Figure 2.1). Running across the top of the screen, right under the application title bar, are the UI elements: the main menu bar, the Status line, and the Shelf. On Mac OS X, note that the main menu bar runs across the top of the screen, above the application title bar.

Figure 2.1 shows the major parts of the UI. In the middle of the interface is the workspace, which is host to your panels (or Scene windows) and their menu options (known as views or viewports in some other 3D packages). This is where most of your focus will be.

Click inside the large Perspective view panel (named *persp*) with the mouse to activate the panel, highlighting its border slightly. Press the spacebar to display a four-panel layout, which gives you top, front, and side views, as well as the perspective view. Press the spacebar in any of the panels to display a large view of that panel.

To the right of the panels is the Attribute Editor/Channel Box/Modeling Toolkit. This is where most of the information (attributes) about a selected object is displayed and edited. Also, this is where you access the Modeling Toolkit suite of polygon tools. Simply click any of the tabs to access these functions. Furthermore, pressing Ctrl+A toggles between the Attribute Editor and the Channel Box.



Time Slider, Range Slider, Playback Controls

In short, the Attribute Editor gives you access to all of an object's attributes, whereas the Channel Box is a quicker display of the most commonly animated attributes of the selected object.

Letter Keys and Syntax in Maya

Maya is case sensitive (meaning it distinguishes between lowercase and uppercase letters). The conventions of this book are to always print an uppercase letter to denote which key you must press. So, when I ask you to press the E key, for example, you should simply press the E key on your keyboard (thereby entering a lowercase e). When an uppercase letter is called for, the book tells you to press Shift+E, thereby entering the uppercase letter *E* into Maya. Also, make sure your Caps Lock key is turned off.

Mouse Controls

Maya requires the use of a three-button mouse, even on a Mac. The clickable scroll wheel found on most mice can be used as the third button by pressing down to click with the wheel.

SHORTCUTS TO NAVIGATING

Here's a rundown of how to navigate Maya. Keep in mind that the Option key is used on a Mac in place of the Alt key on a PC.

Alt+MMB+Click Tracking moves left, right, up, or down in two dimensions; hold down the Alt key, press and hold the MMB, and drag the mouse.

Alt+RMB+Click This dollies into or out of a view, essentially zooming the view in and out. Hold down the Alt key, press and hold the RMB, and drag the mouse.

Scroll Wheel The scroll wheel acts as a middle mouse button when pressed and can also dolly into or out of a view just like the Alt+RMB+click combination when scrolling the wheel.

Alt+Click This rotates or orbits the camera around in a Perspective window. To orbit, hold down the Alt key and the LMB. You can't tumble your view in an orthographic panel.

Alt+Ctrl+Click and Drag Dollies your view into the screen area specified in your mouse drag. Hold down the Alt and Ctrl keys while using the LMB to outline a window in the panel to execute this bounding box dolly. This action is commonly referred to as a window zoom in other applications.

The ViewCube The ViewCube is a navigational aid that may not be visible by default when you launch Maya 2015 due to a change in the default display technology in this new version of Maya. The ViewCube lets you easily change your current panel view, but first, the panel needs to be set to Legacy mode: to enable the ViewCube in a viewport, select Renderer from the panel's menu bar and select either Legacy Default Viewport or Legacy High Quality Viewport, and the ViewCube will appear in the upper right corner of the panel.

By clicking an area of the ViewCube (shown here), you can switch to other views inside that panel. Clicking one of the conical axis markers gives you an orthogonal view from that direction. Clicking the center square gives you the perspective view. You can toggle the ViewCube on or off in the UI by choosing Display → Heads Up Display → ViewCube.



Mac Keys The Option key on a Mac serves the same function as the Alt key on a PC. Although a few Ctrl key combinations in Windows are accessed via the Command key on a Mac, Mac users can use the Mac's Ctrl key for their key combinations just like PC users do.

In Maya, you press and hold the Alt key on a PC (or the Option key on a Mac) along with the appropriate mouse button to move in the view panel.

• The left mouse button (LMB) acts as the primary selection button and allows you to orbit around objects when used with the Alt key.

- The right mouse button (RMB) activates numerous shortcut menus and lets you zoom when used with the Alt key.
- The middle mouse button (MMB) used with the Alt key lets you move within the Maya interface panels, and the mouse's wheel can be used to zoom in and out as well.

Making Selections

Selecting objects in a view panel is as easy as clicking them. As you select an object, its attributes appear in the Attribute Editor or Channel Box on the right. To select multiple objects, simply hold the Shift key as you click objects to add to your current selection. If you press Ctrl+LMB (press the Ctrl key and click) on an active object, you'll deselect it. To clear all of your current selections, click anywhere in the empty areas of the view panel.

Remember: Use Shift+click to select, and Maya adds to the current selection. Use Ctrl+click, and Maya deselects the object you clicked.

Manipulating Objects

When you select an object and enable one of the transformation tools (tools that allow you to move, rotate, or scale an object), you'll see a manipulator appear at or around the selected object. Figure 2.2 shows the three distinct and most common manipulators for all objects in Maya (Move, Rotate, and Scale) as well as the Universal Manipulator. You use these manipulators to adjust attributes of the objects visually and in real time.

To activate a transform tool, select an object and then click one of the transform tool icons in the Tool Box, shown in Figure 2.3.

Press 4 for Wireframe mode; press 5 for Shaded mode.

Try This Let's put some of this into action.

1. Choose Create → NURBS Primitives → Sphere. Drag in a view panel anywhere on its grid to create a wireframe sphere and then size it to your liking. In one of the view panels, press the 5 key on your keyboard, and the display of the sphere will become solid gray. This is called *Shaded mode*. Press the 4 key to return to Wireframe mode. You should also see an in-view message at the top of your view stating "shaded display is now on. Press 4 to display objects in wireframe mode."

You can turn off in-view messages by toggling off the In-View Messages check box under Display → Heads Up Display.

Figure 2.2 The Maya manipulators

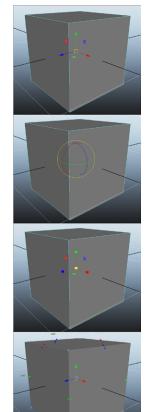


Figure 2.3
The transform tools in the Tool Box



- 2. With the sphere selected, select the Move tool () from the Tool Box. The first manipulator shown in Figure 2.2 should appear in the middle of the sphere. The three arrows represent the three axes of possible movement for the object. Red is for the X-axis, green is for the Y-axis, and blue is for the Z-axis. Cyan is for free movement in both axes of the active panel view. Clicking any one of the three arrows lets you move the object only on that particular axis. The square in the middle of the manipulator lets you move the object freely around the plane of the view panel, regardless of the axis. The three squares represent planar movement in two of the three axes at a time, for example moving on the XY plane or the YZ plane.
- 3. Next, select the Rotate tool () from the Tool Box, and you'll see the second manipulator in Figure 2.2. The three ent the three axes of rotation for the object—red for X green for

colored circles represent the three axes of rotation for the object—red for *X*, green for *Y*, and blue for *Z*. Select a circle to rotate the object on that axis. The yellow circle surrounding the three axis circles lets you freely rotate the object on all three axes.

- 4. Try selecting the Scale tool (to see the third manipulator in Figure 2.2. By selecting one of the axis handles and dragging the mouse, you can scale the object in a non-uniform manner in that axis. The middle cyan box scales the object uniformly on all three axes.
- 5. Try selecting the Universal Manipulator, which is not shown in the Tool Box but is found by choosing Modify → Transformation Tools → Universal Manipulator. Its icon () appears under the Tool Box after you select it from the menu. This tool acts in place of all three manipulators you just tried. Grabbing the familiar arrows translates the sphere. Selecting any of the curved arrows in the middle of the edges of the manipulator box lets you rotate the sphere in that axis. Finally, selecting and dragging the cyan boxes in the corners of the manipulator box lets you scale the sphere. If you hold down the Ctrl key as you drag, you can scale the sphere in just one axis.

Go ahead and click around the interface some more. Create more primitive objects and tool around a bit. Move around the view panels and see how it feels. Give the tires a good kick. Enough chatting—let's jump into the solar system exercise.

Project: The Solar System

This project will familiarize you with the fundamentals of navigating Maya, object creation, hierarchy, and pivots, all of which are important concepts for scene manipulation and animation within Maya. In this exercise, you'll gain experience with UI elements while setting up a Maya project in which you create and manipulate objects. You will

animate a simple simulation of your working solar system, making and adjusting hierarchies for animation and setting keyframes.

Starting with the Sun in the center, the planets in order are Mercury, Venus, Earth, Mars, Jupiter, Saturn, Uranus, Neptune, and Pluto. (Yes, yes, I know Pluto isn't classified as a planet anymore.) All these planets orbit the Sun in ellipses, but you'll give them circular orbits for this exercise. Most planets have a number of moons that orbit them, and a few, including Saturn, have large rings that circle them. For this exercise, we will create only two moons for any planet that has more than two moons, like Jupiter.

The more you run this exercise, the clearer the scene manipulation and hierarchy structure will become to you. Art is a marriage of inspiration, hard work, and practice.

Creating a Project

Projects are the way Maya manages a scene's assets. A file and folder structure keeps your files organized according to projects. You will want to have a new project for this new exercise.

The top level of this organization is the *project folder*. Within the project folder are numerous file folders that hold your files. When you set your workspace to a project folder (or when you create a new project), Maya will know where to look for elements and folders for that project. The two most important types are the Scenes and Images folders. The Scenes folder stores your scene files, which contain all the information for your scene. The Images folder stores images you've rendered from your scene.

NAMING OBJECTS AND KEEPING THE SCENE ORGANIZED

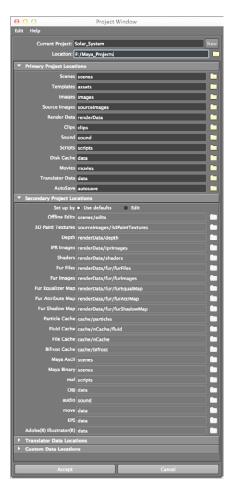
In Maya and most other CG packages, keeping things organized is important to workflow. When you pick up a disorganized scene, it is very time-consuming to figure out exactly how everything works together. Get into the habit of naming your objects and keeping a clean scene, or you'll be bombarded by dirty looks from other artists when they have to handle your cluttered scenes.

The scene files discussed in this chapter are included in the book's download, available at www.sybex.com/go/introducingmaya2015. They are in a project layout explained in the following text. Copy the scene files on the web page for this project into your own project folders after you create the project.

To create a new project for this assignment, follow these steps:

1. Choose File → Project Window to open the Project Window and then click the New button.

Figure 2.4 The completed **Project Window**



2. In the Current Project field in the Project Window, enter **Solar_System** as the name for your project. In the Location box, type the location where you want to store your projects.

The default location for Windows XP. Vista, and Windows 7 is the current user's My Documents folder: My Documents maya\projects; for Macs, the default location is Home (/Users/<yourname>) in the Documents/maya/projects/default folder; and for Windows 8 users it is found under the Documents location under the Start menu. If you prefer, you can put projects in a folder on a secondary or external hard drive to keep them separate from your operating system; this allows for easier backup and is generally a safer environment.

3. If you're using a Windows system, create a folder on your hard drive called **Projects** using Windows Explorer. If you're using a Mac, select a drive in the Finder and create a folder on the drive called **Projects**. In the Project Window, click the folder icon next to the Location field and select

D:\Projects (Windows) or <Hard Drive Name>/Projects (Mac) for the location. Maya will fill in all the other fields for you with defaults. Click Accept to create the necessary folders in your specified location. Figure 2.4 shows the completed Project Window in Mac OS X; except for the drive name, the values are the same on Windows.

After you create projects, you can switch between them by choosing File → Set Project and selecting the new project. Maya will then use that project's folders until you switch to or create another project. You may also select a recent project by choosing File → Recent Projects.

DON'T FORGET TO ALWAYS SET YOUR PROJECT FIRST!

You should make sure to set your project before continuing with your work. The exercises in this book are based on projects, and you'll need to set your project whenever you start a new exercise. Otherwise, the scene may not load properly, or your files may not save to the proper locations for that project. Don't say I didn't warn you!

The Production Process: Creating and Animating the Objects

In this project, you'll first create the Sun, the planets, and their moons; then, you'll animate their respective orbits and rotations.

USE CUBES INSTEAD OF SPHERES

Make the planets with cubes instead to make it easier to see them spin.

Creating the Sun and the Planets

The first thing you'll do is create the Sun and the planets. Follow these steps:

- 1. Choose File → New Scene (or press Ctrl+N). Maya asks if you want to save your current scene. Save the file if you need to or click Don't Save to discard the scene.
- 2. By default, the screen should begin in an expanded perspective view. Press the spacebar to enable the four-panel view. When you're in the four-panel view, press the spacebar with the cursor inside the top view panel to select and maximize it.
- 3. To create the Sun, you need to create a primitive sphere. A *primitive* is a basic 3D shape. First let's turn off a default feature called Interactive Creation. Uncheck Create → NURBS Primitives → Interactive Creation to toggle it off, as shown in Figure 2.5.

Interactive Creation lets you create a primitive by clicking and dragging to specify its size and position. You'll find this option when you choose Create → NURBS Primitives or Create → Polygon Primitives. When this option, at the bottom of each of those menus, is unselected or off, the created primitive appears at the origin in 3D space at a uniform scale of 1.0.

- 4. With Interactive Creation now turned off, choose Create → NURBS Primitives → Sphere. Doing so places a NURBS sphere exactly at the origin—that is, at a position of 0,0,0 for *X*,*Y*,*Z*. This is good, because the origin of the work
 - space will be the center of the solar system, too.
- 5. Select the word *nurbsSphere1* in the Channel Box to the right of the Maya UI (shown in Figure 2.6) and enter **Sun** to rename it. If you don't see the Channel Box in your Maya window, press Ctrl+A to toggle the view from the Attribute Editor to the Channel Box.

Figure 2.5 Turning off Interactive Creation

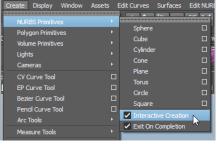
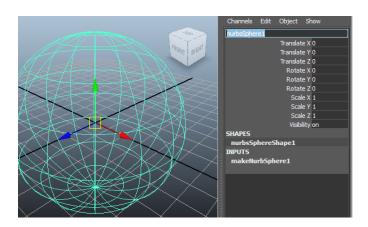


Figure 2.6 Renaming the sphere in the **Channel Box**



Naming your objects right after creation is a good habit to develop and is particularly important if anyone needs to alter your scene file; proper naming will keep them from getting frustrated when they work on your scene.

Always keep in mind that Maya is case sensitive. An object named "sun" is different from an object named "Sun."

- 6. Choose the Scale tool in the Tool Box to activate the Scale manipulator, and uniformly scale the Sun sphere up to about four times its creation scale of 1. (Make it a scale of about 4 in all three axes.) For more precision, you have two options: You can select the sphere and enter a value of 4 in all three entry fields (the white window next to the attribute) for the Scale X, Scale Y, and Scale Z channels in the Channel Box shown on the top in Figure 2.7. Alternatively, you can enter a value of 4 in the Input box on the Status line along the top of the UI, as shown on the bottom of Figure 2.7.
- 7. After you enter the final value through either method, press Enter, and the sphere will grow to be four times its original size. Entering exact values in the Channel Box or Input box is a way to scale the sphere precisely; using the manipulator isn't as precise.



The Sun sphere's Scale values in the Channel Box (above) and in the Input box (below)

Creating the Planets

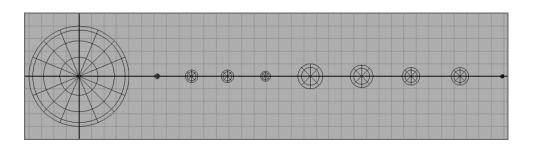
Next, you'll create the primitive spheres for the planets. Leave Interactive Creation off, and follow these steps:

- 1. Create a NURBS sphere for Mercury just as you did before, by choosing Create → NURBS Primitives → Sphere. A new sphere appears at the origin. Click its name in the Channel Box and change it to Mercury.
- 2. Choose the Move tool from the Tool Box to activate the Move manipulator and move Mercury a few grid units away from the Sun sphere in the positive *X* direction. (Click

- the red arrow and drag it to the right.) Leave about 2 grid units between the Mercury sphere and the Sun sphere.
- 3. Because Mercury is the second-smallest planet and is tiny compared to the Sun, scale it down to 1/20 the size of the Sun sphere, or type 0.2 in all three axes of Scale in the Channel Box.
- 4. Repeat steps 1 through 3 to create the rest of the planets and line them up, placing each one farther out along the *X*-axis with about two grid units of space between them. Scale each one proportionally as shown in the chart on the right.

These proportions aren't exactly real, but they will do nicely here. Figure 2.8 shows how your solar system should look now.

No, Pluto isn't actually a planet anymore, but for nostalgia's sake, we'll include it here in our solar system. Poor Pluto!



PLANET SPHERE SIZES Venus 0.5 Earth 0.5 Mars 0.4 **Jupiter** 1.0 0.9 Saturn Uranus 0.7 Neptune 0.7 Pluto 0.15

Figure 2.8 All the NURBS spheres are lined up in place (top view).

Using Snaps

Now is the perfect time to start using *snaps*. Table 2.1 shows some common snap icons. These icons run across the top of the UI just below the main menu bar, as shown here.

SNAP ICON	NAME	DESCRIPTION	Table 2.1
1	Snap To Grids	Snaps objects to intersections of the view's grid	Snap icons
©	Snap To Curves	Snaps objects along a curve	
•	Snap To Points	Snaps objects to object points such as CVs or vertices	
*	Snap To Projected Center	Snaps an object to the center of another object	
%	Snap To View Planes	Snaps objects to view planes	

You use snaps to snap objects into place with precision, by placing them by their pivot points directly onto grid points, onto other object pivots, onto curve points, and so on. Here you'll reposition all the planets slightly to center them on the nearest grid line intersection:

- 1. Select the first planet, Mercury. Choose the Move tool from the Tool Box and toggle on grid snaps by clicking the Snap To Grids icon (1811).
- 2. The center of the Move manipulator changes from a square to a circle, signaling that some form of snapping is active. Grab the manipulator in the middle by this circle and move it slightly to the left or right to snap it onto the closest grid intersection on the X-axis.
- 3. Select the remaining planets and snap them all to the closest grid intersection on the X-axis, making sure to keep about two grid spaces between them. Because the Sun sphere was created at the origin and you haven't moved it, you don't need to snap it onto an intersection.

Making Saturn's Ring

Now, create the ring for Saturn. To do so, follow these steps:

- Press the spacebar in the top view to maximize it. Choose Create → NURBS Primitives → Torus to place a donut shape at the origin (since Interactive Creation is still off). Use the Move tool to snap the ring to the same grid intersection as Saturn. This ensures that both the planet and its ring are on the same pivot point and share the same center.
- 2. Select the torus shape you've created and name it **Ring** in the Channel Box.
- 3. While the torus shape is still selected in the top view, press the spacebar to display the four-panel layout. Place the mouse cursor in the persp view and press the spacebar to maximize the Perspective window.
- 4. Press the F key to focus the perspective display on the ring and on Saturn. Pressing F centers and zooms in the panel on just the selected object(s).
- 5. Press 5 to get into Shaded mode.
- 6. From the Tool Box, select the Scale tool and scale the torus down to 0 or close to 0 in the *Y*-axis (the torus's height, in this case) to flatten it.
 - You need to edit the attributes of the ring to increase the inside radius of the donut shape and create a gap between the planet and the ring.
- 7. Press Ctrl+A (Ctrl/Cmd+A will work on a Mac) to toggle the Attribute Editor if it's not on and then click the makeNurbTorus1 tab to select its creation node (see Figure 2.9).
- 8. Increase the Radius attribute to about 1.5 and decrease the Height Ratio attribute to about 0.25 to get the desired effect.

Now all your planets are complete, and you can move on to the moons.

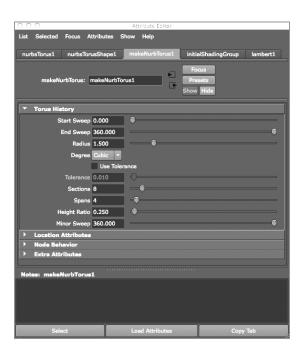


Figure 2.9 Changing the creation attributes of the NURBS torus in

the Attribute Editor

Saving Your Work

Save your work, unless you like to live on the edge. Power failures and other unforeseen circumstances (such as your pet jumping onto your keyboard) may not happen often, but they do happen. Because you created this as a new project, the Save File window will direct you to the Scenes folder of that project. Save your scene as planets in the .mb (Maya Binary) format.

The file Planets_v1.mb in the Scenes folder of the Solar_System project, available on the book's web page, shows what the scene should look like at this point.

SAVING MULTIPLE VERSIONS OF YOUR WORK

Maya software's Incremental Save feature makes a backup of your scene file every time you save your scene with File → Save. To enable it, choose File → Save Scene □ and click the Incremental Save box. Every time you save, Maya will create a new backup (for example, planets_001.mb) in a folder in your Scenes folder.

The scene files for the projects in this book use a slightly different naming system than the names generated by Incremental Save so there is no risk of files overwriting each other.

The AutoSave feature automatically saves your file at a set interval of time. To enable AutoSave, select Window → Settings/Preferences → Preferences. In the Categories column on the left, select Settings → Files/Projects. You can enable AutoSave as well as set its time interval and save location under the AutoSave heading.

Creating the Moons

For the planets with moons, create a new NURBS sphere for each moon. For simplicity's sake, create only two moons for any planet. The first moon will be Earth's. Use the top view to follow these steps:

- 1. Create a NURBS sphere and scale it to about half the size of Earth using the Scale tool. Visually estimate the size of the moon.
- 2. Move the sphere to within half a unit of Earth, using the Move tool by the *X*-axis. There's no need to snap it to a grid point, so toggle off the Snap To Grids icon (1811). Name the moon earthMoon.
- 3. Repeat steps 1 and 2 for the remaining moons, placing them each within half a grid unit from their respective planets. When placing two moons, place them on opposite sides of the planet. Make sure to name your moons.
- 4. After you've finished with all the moons, their placements, and their sizes, you should have a scene similar to Figure 2.10 in perspective view. If you don't, it's clear Maya doesn't like you.

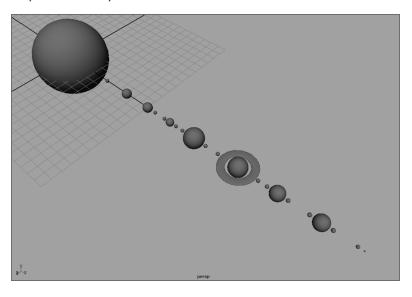


Figure 2.10

The planets and moons in position in perspective view

Applying a Simple Shader

To help distinguish one gray planet from another, let's attach simple shaders to each of the planets to give them color. Shaders, in short, are materials that give an object its particular look, whether it is color or a tactile texture. Follow these steps:

1. Choose Window → Rendering Editors → Hypershade to open the Hypershade window. With this window, you create the look of your objects by assigning colors, surface properties, and so on. You'll notice three default (or initial) shader icons already loaded (see Figure 2.11).



Figure 2.11 The Hypershade window

- 2. In the Create Maya Nodes panel on the left of the Hypershade window and under the Surface heading, click the Lambert icon (a gray sphere) to create a new Lambert shader node. It appears in the top and bottom of the Hypershade window. Click this icon eight more times to create a total of nine Lambert shading groups in the Hypershade window.
- 3. Click the first of the new Lambert nodes (lambert2) in the Hypershade window to display its attributes in the Attribute Editor on the right of the UI. If you have the Channel Box displayed instead, double-click the shader's icon to open the Attribute Editor. At the top, rename lambert2 as Mercury_Color to identify this material as the one you'll use for Mercury.
- 4. Name each of the remaining planets in your animation (Venus, Earth, Mars, Jupiter, Saturn, Uranus, Neptune, and Pluto) and save your work.

After you've created and named all the shaders, you can assign appropriate colors to each of the shaders according to the planets they represent.

- 1. Double-click Mercury's thumbnail icon to open the shader's Attribute Editor (see Figure 2.12).
- 2. To change the color of the shader, click the gray box next to the Color attribute. This opens the Color Chooser window, where you can choose a new color from the color

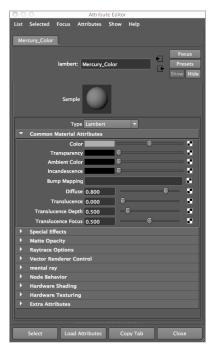


Figure 2.12

Mercury's shading group in the **Attribute Editor**

Figure 2.13

The Color Chooser window

wheel or by adjusting values with the HSV sliders. Go with an orange color, such as in Figure 2.13 (take note of the HSV values).



3. Change the remainder of the shaders as follows:

Mercury	Orange-brown
Venus	Beige-yellow
Earth	Blue
Mars	Red-orange
Jupiter	Yellow-green
Saturn	Pale yellow
Uranus	Cyan
Neptune	Aqua blue
Pluto	Bright gray

Figure 2.14 shows the shading groups.

4. Next, apply shaders to the planets. Select a planet in the Perspective window, and RMB+click its corresponding material in the Hypershade window to open a marking menu. Marking menus are a fast UI workflow to allow you to select commands and options as you work in your panels without having access the main menu bar.



Figure 2.14

The Hypershade window with all the colored planet shading groups 5. Still holding the right mouse button, drag up to highlight Assign Material To Selection and release the button to select it. You can also use the middle mouse button to drag the material from the Hypershade window to its planet. Leave the moons set to the default gray color. When you're finished, you should have a scene similar to Figure 2.15.

A marking menu is a context-sensitive menu that appears when you RMB+click an object or a shader. For example, clicking a polygon cube in a scene brings up a marking menu where you can select the cube's vertices, faces, edges, and so on, and where you can quickly access some often-used options and tools. The kind of marking menu that appears depends on the node you are RMB+clicking.

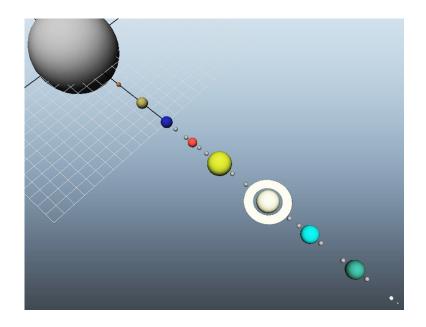


Figure 2.15 The shaded planets in perspective view

Now that you're finished, you're ready to animate. Save this file; if you enabled Incremental Save as recommended earlier, your file won't be replaced with subsequent saves.

Creating the Animation

To begin this phase of the project, load the file Planets_v2.mb in the Scenes folder of the Solar_System project on the web page to your hard drive, or continue with your own scene file.

The animation you'll do for the planet's self-rotation is straightforward. You'll rotate the planets around their own axes for their self-rotation, and then you'll animate the moons around the planets for their lunar orbits. Finally you'll make the planets and their moons orbit the Sun.

This exercise focuses on hierarchy and pivot points. A pivot point is an object's center of balance of sorts. Every object or node created in Maya has a pivot point, especially primitives like you're creating here. That pivot point is usually at the center of the primitive and moves along with the object if it's moved.

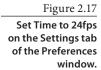
Now, you need to set up the animation settings for your scene file:

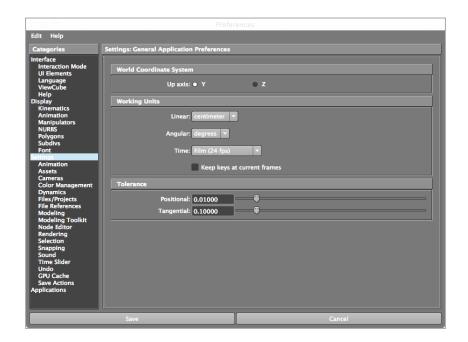
- 1. Press F2 to open the Animation menu set, or select it from the drop-down menu shown earlier in the chapter. *Menu sets* are groupings of menu headings in the main menu bar. They're organized according to the type of task at hand. You'll see the several menu headings change.
- 2. At the bottom of the UI, you'll notice a slider bar (the Range slider) directly below the strip of numbers counting off the frames (the Time slider) in the scene. Using the Range slider, you'll set the length of your animation to go from 1 to 240. Enter 1 in the Scene Start Frame and Range Start Frame boxes (Figure 2.16). Enter a value of 240 in the Scene End Frame and Range End Frame boxes, also as shown in Figure 2.16.

Figure 2.16 The Time and Range sliders



3. To the right of the Range slider, click the Animation Preferences icon (), click Settings (Figure 2.17), and make sure Time is set to Film (24fps).





Choose Window → Settings/Preferences → Preferences to open the Preferences window. Under Settings → Undo, make sure Undo is on and set Queue to Infinite, allowing you to press Ctrl-Z as many times as needed. To close the Preferences window, click Save.

Mercury's Rotation

Now you're ready to animate Mercury's rotation. Follow these steps:

- 1. Select Mercury, and press E to activate the Rotate tool. The E key is the hotkey to invoke the Rotate tool in Maya. Press F to focus on Mercury in the perspective view, or zoom in on it manually.
- 2. Make sure you're on frame 1 of your animation range by clicking and dragging the Scrub bar (refer to the earlier Figure 2.17) to place it at the desired frame. You can also manually type the frame value of 1 in the Current Frame box.
- 3. For Mercury, you'll set your initial keyframe for the Y-axis rotation. RMB+click the attribute name for Rotate Y in the Channel Box to select it and bring up a context menu as shown in Figure 2.18. Select Key Selected from the context menu. This places a keyframe for a rotation of 0 in the Y-axis at frame 1 for the Mercury sphere, and the Value box in the Rotate Y attribute turns to pink to indicate that a keyframe exists for that attribute.

Conversely, instead of RMB+clicking Rotate Y in the Channel Box, you could also set a keyframe through the main menu bar (in the Animation menu set) by selecting Animate → Set Key. However, this method will set a keyframe for *all* attributes of the Mercury sphere, which is not efficient. You should ideally try to set keyframes only on attributes you need.

- 4. Using the Scrub bar in the Time slider, go to frame 240. Grab the Rotation manipulator handle by the Y-axis (the green circle) and turn it clockwise a few times to rotate the sphere clockwise. You'll notice that you can rotate the object only so far in one direction before it seems to reset to its original rotation. Rotate it as far as it will go and release the mouse button. Then, click the manipulator again and drag to rotate the sphere as many times as necessary until you're satisfied.
- 5. RMB+click the Rotate Y attribute in the Channel Box and choose Key Selected. This sets a keyframe for your new *Y*-axis rotation at frame 240 for the Mercury sphere.
- 6. To play back your animation, you can scrub your Time slider. Click in the Time slider on the Scrub bar, hold down the left mouse button, and move your cursor from side to side to scrub in real time. You'll see Mercury rotating in your active view panel, if you set your two keyframes as described.

Figure 2.18 Setting the initial keyframe for Mercury's Y-axis rotation



You have the self-rotation for Mercury worked out. Mercury has no moon, so let's get Mercury orbiting the Sun.

Grouping Mercury for a New Pivot Point

You've learned that every object in Maya is created with a pivot point around which it rotates and scales and which acts as the placement for its X-, Y-, and Z-coordinates. To orbit Mercury around the Sun sphere, the sphere must revolve around a pivot point that is placed in the middle of the Sun sphere. If the pivot point for Mercury is already at the center of itself, how can you revolve it around the Sun sphere?

One idea is to move its current pivot point from the center of itself to the center of the Sun sphere. That would, however, negate Mercury's own rotation, and it would no longer spin around its own center, so you can't do that. You need to create a new pivot point for this object. This way, you have the original pivot point at Mercury's center so it can selfrotate, and you have a second pivot point at the Sun sphere so that Mercury can revolve around that point around the Sun sphere. You'll accomplish this by creating a new parent node above Mercury in the hierarchy. What does that mean?

To explain, I'll take time in the following section to introduce the concept of Maya object structure: nodes and hierarchies. Save your progress so far and open a new blank scene. After this explanation, you'll resume the solar system exercise.

Hierarchy and Maya Object Structure

On top of everything you see in Maya (its interface) is a layer you don't see: the code. The layer of code keeps the objects in Maya organized through a network of nodes. How you relate these nodes defines how you've built your scene. So, having a solid understanding of how Maya defines objects and how they interact is essential to an efficient and successful animation process.

Understanding Nodes

At its core, Maya relies on packets of information called *nodes*, and each node carries with it a group of attributes that in combination define an object. These attributes can be spatial coordinates, geometric descriptors, color values, and so on. You can define, animate, and interconnect any or all of these attributes individually or in concert, which gives you amazing control over a scene.

Nodes that define the shape of a surface or a primitive are called *creation nodes* or shape nodes. These nodes carry the information that defines how that object is created. For example, a sphere's creation node has an attribute for its radius. Changing that attribute changes the radius of the sphere at its base level, making it a bigger or smaller sphere. This is different from scaling the sphere as you've done with the planets so far. Shape nodes are low on the hierarchy chain and are always child nodes of transform nodes. The

sphere listens to its creation node attributes first and then moves down the chain to its other nodes' attributes (such as position, rotation, or scale).

The most visible and used nodes are the transform nodes, also known as directed acyclic graph (DAG) nodes. These nodes contain all the transformation attributes for an object or a group of objects below it. Transformations are the values for translation (position), rotation, and scale. These nodes also hold hierarchy information about any other children or parent nodes to which they're attached. When you move or scale an object, you adjust attributes in this node.

Parents and Children

A parent node is simply a node that passes its transformations down the hierarchy chain to its children. A child node inherits the transforms of all the parents above it. So, by using hierarchies for the solar system exercise, you'll create a nested hierarchy of parents and children to animate the orbital rotation of the nine planets and some of their moons.

With the proper hierarchy, the animation of the planet (the parent) orbiting the Sun automatically translates to the moon (the child). In effect, the planet takes the moon with it as it goes around the Sun.

Child nodes have their own transformations that can be coupled with any inherited transforms from their parent, and these transformations affect them and any of their children down the line.

You're about to experience this first hand as you continue the solar system exercise. The more you hear about these concepts in different contexts, the easier they will be to master.

Figure 2.19 shows the Outliner and Hypergraph views with a simple hierarchy of objects for your reference. The Outliner and Hypergraph show you the objects in your scene in an outline and flowchart format, respectively. Both of these windows allow you to access the different levels of nodes (the hierarchy) in a scene and are discussed further in Chapter 3.

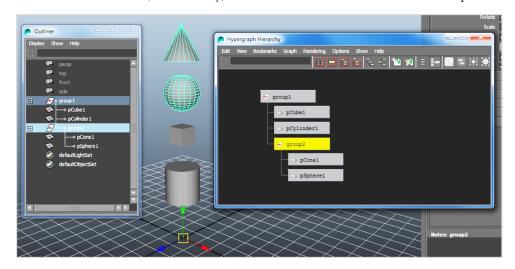


Figure 2.19

A simple hierarchy in both the Outliner and Hypergraph windows

A top parent node called group1 holds its children pCube1, pCylinder1, and the nested group node group2. The node group2 is the parent node of pCone1 and pSphere1.

The Solar System, Resumed

If you still feel a little unsure about nodes and hierarchies, take the time to reread the previous section. You'll practice these concepts as you resume the solar system exercise. By the time you've finished this exercise, you'll have a strong sense of how hierarchies work in Maya, although you should feel free to repeat the entire exercise if you think that will help you master hierarchies. Understanding nodes and hierarchies is absolutely critical to animating in Maya.

If you're new to CG animation, take your time with the following section.

Animating Mercury's Orbit Around the Sun

Load your scene from where you last saved it. When you left off, you had created the selfrotation animation for Mercury and were about to create a second pivot point for the planet to orbit around the Sun sphere by creating a new parent node for the Mercury sphere.

To create a new pivot point by making a new parent node, follow these steps:

1. With Mercury selected, press E for the Rotate tool and then choose Edit → Group from the main menu bar. The Channel Box displays attributes for a new node called group1. Notice that nothing about the Mercury sphere changed, except that the Rotation manipulator handle jumped from where it was originally centered on Mercury all the way back to the origin, where the zero points of the X-, Y-, and Z-axes collide. Figure 2.20 shows the new Mercury group and its new pivot location.

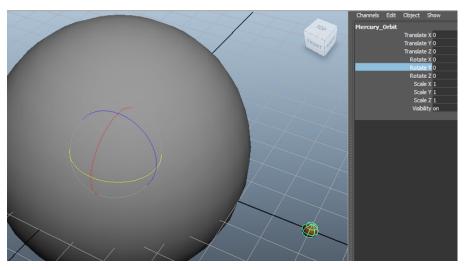


Figure 2.20 Grouping Mercury to itself creates a new pivot point at the origin.

You just created a new Maya object by grouping Mercury to itself. In doing so, you also created a second pivot point for Mercury, which was placed by Maya at the origin by default. Because an object's manipulator always appears at the object's pivot point when it's selected, Mercury's Rotate manipulator jumped to the origin when the new parent node was created. That is fortunate for you because that happens to be the center of the Sun sphere—exactly where you need it to be for Mercury to orbit the Sun sphere properly.

- 2. Without unselecting Mercury, click group1 in the Channel Box, and change the name of this new group to Mercury_Orbit. It's important to make the distinction between node names so you never get confused. Now you know that the Mercury node is the planet sphere itself, whereas Mercury_Orbit is the name of the new parent node with which you'll orbit Mercury around the Sun sphere.
- 3. Click anywhere in an empty space in your view window to unselect Mercury_Orbit. Try selecting it again by clicking the Mercury sphere. Notice that when you click Mercury, you select only the planet and not the new parent node Mercury_Orbit, the group that has its pivot point at the center of the Sun sphere. This happens because you're in Object mode. To select the group Mercury_Orbit, switch into Hierarchy mode by toggling its icon () on the Status line at the top of the UI, as shown in Figure 2.21. For more on selection modes, see Chapter 3. Now when you click Mercury, it selects the new Mercury_Orbit parent node you made.
- 4. Go back to frame 1 of your animation. Set a keyframe for Mercury_Orbit's Rotate Y attribute by RMB+clicking its name in the Channel Box and then choosing Key Selected.
- 5. Go to frame 240, grab Mercury_Orbit's Rotate Manipulator handle by the green Y-axis, and spin it around the Sun twice in either direction. (It doesn't matter if you go clockwise or counterclockwise.) You can also enter 720 in the Rotate Y attribute field in the Channel Box (or -720 to go in the other direction).
- 6. Set a keyframe at frame 240 for Mercury_Orbit's Rotate Y attribute, as you did in step 4. Scrub your animation to play it back.
- Make sure to switch back to Object mode by clicking its icon () in the Status line (Figure 2.21).

Does that make good sense? You'll have the chance to do this a few more times as you animate the other planets and their moons. However, if you still find yourself a little fuzzy on this concept (which is perfectly normal), repeat the steps to animate Mercury in a new scene file if need be. One down, eight to go.

Creating Venus

For your next planet, Venus, follow the same procedure as for Mercury's self-rotation (steps 1-6 in the "Mercury's Rotation" section) and animate it so that it rotates around itself, just like Mercury. Then, create a new pivot point by grouping Venus to itself to

Figure 2.21 Toggling on the Hierarchy mode



create a new parent node (as you did for Mercury_Orbit in steps 1-2 in the previous section) and call the new parent node **Venus_Orbit**. Lastly, animate Venus_Orbit to revolve around the Sun sphere just as you did with Mercury_Orbit in steps 3-7.

Earth and the Moon

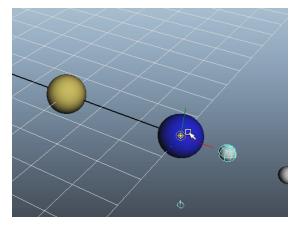
Now you need to animate the third planet, Earth, in much the same way, except that this time there will be the added complication of a moon. (Earth? Hey, I can see my house from here!)

To animate Earth and the moon, follow these steps:

- 1. Select Earth and give it its self-rotation animation as you did for Mercury.
- 2. Select the moon and give it its own self-rotation animation by spinning it around itself and keyframing it as you've just done with Earth.
- 3. To spin the moon around Earth, you'll do as you did earlier in this chapter to orbit a planet around the Sun: first, group the moon to itself by choosing Edit → Group; then, name the new parent node **Moon_Orbit**.

This time, however, you need the pivot point to be at the center of Earth and not at the center of the Sun object, where it is currently. Follow these steps:

- 1. Turn on the grid snap (🔝) and press and hold down the D key to be able to move the pivot point. The moon's manipulator changes from a rotation handle to the Pivot Point manipulator. This manipulator acts just like the Move manipulator, but instead of moving the object, it moves the object's pivot point as long as you keep the D key pressed.
- 2. Grab the yellow circle in the middle of the manipulator and move the pivot point to snap it to the grid point located at the center of Earth (see Figure 2.22).



3. Now release the D key to return to the Rotation manipulator for Moon_Orbit. At frame 1, set a keyframe for the moon's Y-axis rotation. Then, at frame 240, rotate the moon around the *Y*-axis and set a keyframe. Return to frame 1.

Figure 2.22 Moving the moon's pivot point to the center of Earth

Grouping the Moon with Earth

To animate Earth's orbit of the Sun, you need to make sure the moon will also follow Earth around the Sun sphere. Instead of just selecting Earth and grouping it to itself as you've done for the other two planets, you need to include the Moon_Orbit node in that group. Follow these steps:

1. Select Earth. Shift+click the Moon Orbit group while in Hierarchy mode () to make sure you get the topmost node of the moon and then choose Edit → Group. Name this new parent node Earth_Orbit. Remember, when you select just Earth or the moon in Object mode (), the Earth_Orbit node isn't selected. If you select Earth and then Shift+click the moon, you select both objects, but you still don't select the parent node Earth_Orbit, which is the group that contains both these objects and has its pivot point at the center of the Sun. Make sure you select the right group. Keep an eye on where the manipulator is when you make your selection. If you have the Earth_Orbit node selected, its manipulator should be in the middle of the Sun sphere. I'll deliberately illustrate this mistake and its consequences when you animate Pluto a little later.

Make sure you use Hierarchy mode (🔥) when you click the moon object to select Moon_ Orbit and not just the moon sphere. Otherwise, you'll lose the animation of the moon orbiting Earth.

- 2. Set a keyframe for Earth_Orbit's Rotate Y attribute at frame 1, again by RMB+clicking Rotate Y in the Channel Box and selecting Key Selected.
- 3. Go to frame 240, spin Earth and the moon around the Sun sphere a few times in whichever direction and for however many revolutions you want, and set a keyframe at frame 240.

Now the first three planets are going around themselves and around the Sun, with a moon for Earth. If you haven't been saving your work, save it now. Just don't save over the unanimated version from earlier.

Creating the Other Planets' Moons

Repeat this animation procedure for the remaining planets and moons, but leave out Pluto for now. (Poor Pluto: first it loses out on being a planet, and now it has to wait for last.)

If you find that one of your moons is left behind by its planet or that it no longer revolves around the planet, you most likely made an error when grouping the moon and planet. Undo (Ctrl+Z) until you're at the point right before you grouped them and try again. If that still doesn't work, start over from the earlier version of the file you saved just before you began animating it. You'll learn how to fix it in the section "Using the Outliner" later in this chapter.

Auto Keyframe

You can also use the Auto Keyframe feature when animating the planets and moons. Auto Keyframe automatically sets a keyframe for any attribute that changes from a previously set keyframe. For example, an initial keyframe for an attribute such as Y-Axis Rotation needs to be set at some point in the animation. The next time the Y-Axis Rotation is changed at a different frame, Maya will set a keyframe at that frame automatically.

To turn on Auto Keyframe, click the Auto Keyframe icon (), which is to the right of the Range slider. When the icon is red, Auto Keyframe is active.

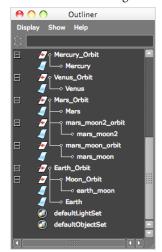
To use Auto Keyframe to animate the moon orbiting Mars, follow these steps:

- 1. Turn on Auto Keyframe.
- 2. Start at frame 1. Select the moon of Mars and set a keyframe for its Y-axis orbit by highlighting Rotate Y in the Channel Box and selecting Key Selected.
- 3. Go to frame 240. Revolve the moon around Mars several times in a direction of your choosing. Maya automatically sets a frame for *Y* rotation at frame 240. Save your file.

Using the Outliner

The Outliner is an outline listing of all the objects and nodes in your scene. For an indepth look at the Outliner, see Chapter 3. For now, let's look at how to use the Outliner to illustrate the hierarchies for the planets and moons. When all is good and proper, the Outliner should look like Figure 2.23. Choose Window → Outliner to open the Outliner window and take a peek at what you have. If you haven't yet properly named everything, including the moons, take this opportunity to do so by double-clicking a name in the Outliner and entering a new name.





Let's look at the planet Mars and its layout in the Outliner to better understand the hierarchy for all the planets. All the other planets should be laid out exactly like Mars (except the planets that have just one or no moon).

At the bottom of the hierarchy are Mars's two moons, mars_moon and mars_moon2. Each of those moons is spinning on its own pivot point. You grouped each moon to itself, created the mars_moon_orbit and mars_moon2_ orbit nodes, and placed their pivot points at the center of Mars to animate their orbits around Mars.

Mars is spinning on its own pivot point, but it needed another pivot point to be able to orbit the Sun. Because you had to make the moons go with it around the Sun, you selected Mars, mars_moon_orbit, and mars_moon2_orbit (the top nodes of the moons that circle the planet Mars) and grouped them all together, placing that pivot point at the center of the Sun. You called this node Mars Orbit. This is the parent node because it's the topmost node for this group. Wherever this parent node goes, the child nodes that are under it will follow.

Hierarchies such as this are a cornerstone of Maya animation. It's imperative that you're comfortable with how they work and how to work with them. If you find yourself scratching your head even a little, try the exercise again. A proper foundation is critical. Remember, this learning 3D thing isn't easy, but patience and repetition help a lot.

Correcting Hierarchy Problems Using the Outliner

One of the most common problems you'll run into with this project is a planet revolving around the Sun without its moon. To illustrate how to fix it using the Outliner, as opposed to undoing and redoing it as suggested earlier, the following steps will force you to make this error with Pluto. Usually, people learn more from mistakes than from doing things correctly.

Go to Pluto, start the same animation procedure as outlined earlier, and then follow these steps to force an error:

- 1. Create Pluto's own self-rotation by spinning it around itself and keyframing as before.
- 2. Do the same for Pluto's moon's self-rotation.
- 3. Group the moon to itself and grid-snap the pivot point at the center of Pluto to create the moon's orbit of Pluto.

When Pluto's moon (pluto_moon) is orbiting Pluto, you're ready to group the moon's orbit and Pluto together to create an orbit of the Sun sphere for both.

- 4. Here is where you make your mistake. In Object mode, select the sphere for Pluto's moon and select the sphere for Pluto. Your error is that you're in Object mode instead of switching to Hierarchy mode.
- 5. Choose Edit → Group to group them together and call that new node **Pluto_Orbit** (following the naming convention you used for the others).
- 6. Animate Pluto_Orbit revolving around the Sun.
- 7. Play back the animation.

Notice that the moon is no longer orbiting the planet. This is because you didn't include pluto_moon_orbit in your group Pluto_Orbit. The animation of the moon going around Pluto is stored in that node, and because it's no longer attached to Pluto_Orbit, there's no moon orbit of Pluto.

Figure 2.24 shows the hierarchy of Pluto and how it's different from that of the other planets: the moon's orbit node has been left out of the group. (Earth has been expanded as a contrasting example.)

Using the Outliner, you can easily fix this problem. Place the pluto_moon_orbit node under the Pluto_Orbit node. Go to frame 1 of the animation, grab the pluto_moon_orbit

Display Show Help 💅 ှ Earth_Orbit Moon_Orbit earth moon pluto_moon_orbit Pluto Orbit faultLightSet

node in the Outliner, and use the middle mouse button to drag it to the Pluto Orbit node so that it has a black horizontal line above and below it to show a connection, as in Figure 2.25.

You've just grouped pluto moon orbit under Pluto Orbit, a practice known as *parenting*. Now you need to parent pluto_moon under pluto_moon_orbit as well. Use the MMB to drag pluto_moon onto pluto_moon_orbit. When you play back the animation, you'll see that the moon is revolving around the planet and that Pluto and the moon are orbiting the Sun sphere. Now that you've corrected Pluto's layout in the Outliner, it's similar to the layouts for the other properly working planets.

The file Planets_v3.mb in the Scenes folder of the Solar System project on the book's web page will give you an idea of how this project should look. The first five planet systems are grouped and animated as a reference, leaving the final four for you to finish.

Figure 2.25 Regrouping objects in the Outliner

Figure 2.24

hierarchy

Pluto's incorrect



You can add objects to a group by MMB+dragging their listing onto the desired parent node in the Outliner. You can also remove objects from a group by MMB+dragging them out of the parent node to a different place in the Outliner.

GROUPING TERMINOLOGY

Grouping terminology can be confusing. Grouping Node A under Node B makes Node A a child of Node B. Node B is now the parent of Node A. Furthermore, any transformation or movement applied to the parent Node B will be inherited by the child Node A.

When you group Node A and Node B, both nodes become siblings under a newly created parent node, Node C. This new node is created just to be the parent of Nodes A and B and is otherwise known as a *null node*. To group objects, select them and choose Edit → Group. Parenting nodes together places the first selected node under the second selected node. For example, if you select Node A, Shift+select Node B, and then choose Edit → Parent, Node A will group under Node B and become its child. This is the same procedure as MMB+dragging Node B to Node A in the Outliner, as you did with Pluto's moon and Pluto itself.

Outputting Your Work: Playblasting

What's the use of animating all this work and not being able to show it? There are several ways of outputting your work in Maya, most of which involve rendering to images. One faster way of outputting your animation in a simple shaded view is called *playblasting*. Playblasting creates a sequence of images that play back on your computer at the proper frame rate. Only if your PC is slow or if you're playblasting a large sequence of frames will your playback degrade. In this case, playblasting 240 frames shouldn't be a problem.

A playblast, as it's called in Maya, outputs the view panel's view into an image sequence or AVI movie. You can also save the image sequence or AVI to disk if you like. Playblasting is done mainly to test the look and animation of a scene, especially when its playback is slow within Maya.

When you have your solar system animated, output a playblast by following these steps:

- 1. With your animation completed, click in the perspective panel to make it active in the four-panel layout (don't maximize the Perspective window). Press 5 to enter Shaded mode.
- 2. RMB+click anywhere in the Time slider, and select Playblast □ from the context menu, as shown in Figure 2.26. The option box is shown for the Playblast options in Figure 2.27.
- 3. In the Playblast options, set Format to gt and Display Size to From Window. Check the Save To File option, and give your playblast a name. Set Scale to 1.0. However, for some users qt is not an option, so you should select avi instead.
- 4. Because you checked the qt option, Maya runs through the animation and creates a QuickTime movie file (or an AVI movie file if you selected the avi option) that is based on the Shaded-mode appearance of the currently active view panel (which should be the Perspective panel). Because you also checked the Save To File option,
 - the movie file is saved to disk. By default, it's saved to the Images folder for the Solar_System project you created on your hard drive. You can also click the Browse button to store the playblast video file anywhere you like. Click the Playblast button.
- 5. When Maya runs through the animation, Windows Media Player or QuickTime automatically opens and plays the movie file of the animation at the proper speed of 24fps, as shown in Figure 2.28. Now you can share your animation with others without having to open Maya and play it back in the scene.

Figure 2.26 Selecting Playblast

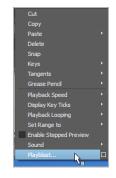


Figure 2.27

The option box for creating a playblast preview

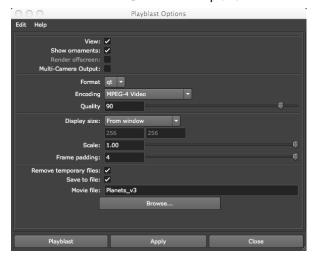
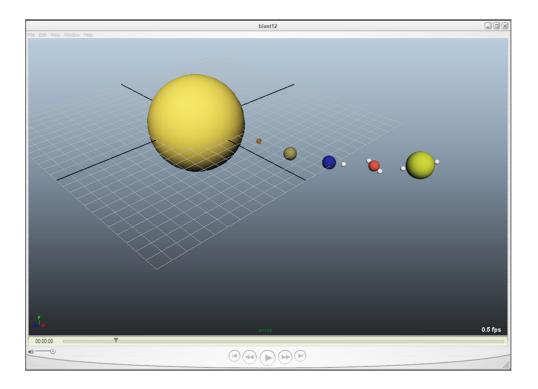


Figure 2.28 Creating a playblast movie file is easy.



Summary

In this chapter, you learned how to start working in Maya by learning how to navigate the UI. Then, you began working by creating a new project, creating basic objects such as primitives, and placing objects in the scene. You learned how to place pivot points for objects and how to use snaps to place points precisely. You had some experience with the Channel Box and Attribute Editor to set an object's attributes. You then went on to create simple shaders for your objects and set keyframes to animate a solar system. You explored object hierarchy and grouping conventions to organize your scene better, and finally you learned how to output a basic playblast video file of your completed animation.

The planet animation you created is based on a system of layering simple actions on top of each other to achieve a more elaborate result. Much of your time in actual animation, as opposed to setup or modeling, will be spent adjusting the small things. These small things give the scene life and character. You'll find that finishing 85 percent of a scene will take about 15 percent of the time. The remaining 85 percent of the time goes into perfecting the final 15 percent of the scene.



The Autodesk® Maya® 2015 Interface

This chapter takes you on a guided tour of all the elements visible on the screen for the Autodesk* Maya* 2015 program as you build a simple model of a decorative box. The chapter draws from the experience you had in Chapter 2, "Jumping into Basic Animation Headfirst," with the solar system exercise. You'll visit the menus, icons, and shelves to become familiar with the interface basics as you build a model.

This chapter also serves as a good reference when you're wondering about the purpose of a particular icon.

Learning Outcomes: In this chapter, you will be able to

- Recognize and use Maya UI elements
- Understand how Maya view panels and windows work
- Use manipulators to transform objects in 3D space
- Create and use reference planes for modeling from pictures
- Use polygon modeling techniques
 - Extrude
 - Bevel
 - Edge loops
 - Interactive Split tool
 - Component editing—edges, faces, vertices
- Use the Layer Editor to organize your scene
- Render test frames to preview your work
- Gain confidence in using the Attribute Editor
- Better manage your scenes and object hierarchies with the Outliner

Navigating in Maya

The key to being a good digital artist or animator isn't knowing where to find all the tools and buttons but knowing how to find the features you need. The purpose of this chapter is to help you get to know Maya and how it operates, building on your experience so far.

Explore the interface. Using your mouse, check out the menus and the tools. Just be careful not to change any settings; the rest of this book and its projects assume your Maya settings are all at their defaults. If you do change some settings inadvertently, reverting to the defaults is easy. Choose Window → Settings/Preferences → Preferences. In the Preferences window, choose Edit → Restore Default Settings. Now all the settings and interface elements are restored to their default states.

Exploring the Maya Layout

Let's take another look at the initial Maya screen in Figure 3.1—this time with the Full Perspective window and not the four-panel layout you saw in the previous chapter.

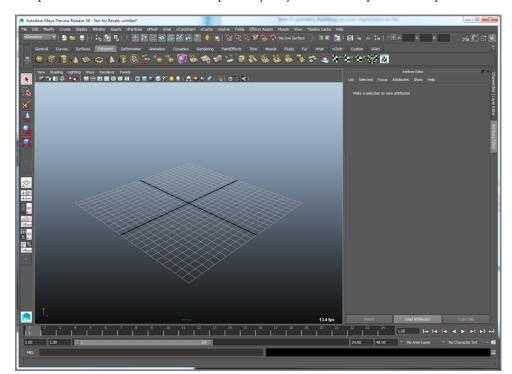


Figure 3.1 The initial Maya screen

> The main menu bar, Status line, and Shelf all run across the top of the screen. The Tool Box runs vertically on the left side of the screen. It contains icons for your transform tools (such as Move, Rotate, and Scale) as well as quick-view selections to allow you to

customize your panel layouts quickly. The Attribute Editor, Channel Box/Layer Editor, and Modeling Toolkit (the Attribute Editor is currently displayed in Figure 3.1) run down the right side of the screen. Finally, listed from the top down, the Time slider, the Range slider, the Character Set menu, the Auto Keyframe button, and the Animation Preferences button, some of which you've already used, run across the bottom of the screen.

REMINDER: MAYA MOUSE CONTROLS

In Maya, holding the Alt key on a PC or the Option key on a Mac along with the appropriate button allows you to move in the view panel. The left mouse button (LMB) acts as the primary selection button and lets you orbit around objects when used with the Alt key. The right mouse button (RMB) activates numerous context menus and lets you zoom with the Alt key. The middle mouse button (MMB) with the Alt key lets you move within the Maya interface, and the mouse's wheel can also be used to zoom in and out.

The Main Menu Bar

In Maya, menu choices are context sensitive; they depend on what you're doing. The main menu bar is shown in Figure 3.2. By switching menu sets, you change your menu choices and hence your available tool set. The menu sets in Maya are Animation, Polygons, Surfaces, Dynamics, Rendering, and nDynamics.



Figure 3.2

The main menu bar is where the magic happens.

Menu Sets

The Menu Set drop-down is the first thing on the Status line, as shown in Figure 3.3.

No matter which menu set you're working in, the first six menu items are constant: File, Edit, Modify, Create, Display, and Window, as is the last menu entry, Help.

Some plug-ins can also add menu items to the main menu bar. If the plug-in is turned off, that menu item is removed. So, don't panic if you don't see the same main menu bar pictured throughout this book.



Figure 3.3

The Menu Set drop-down menu

ADVANCED TIP: FLOATING MENUS In Maya, you can tear off menus to create separate floating boxes that you can place anywhere in the workspace, as shown here. This makes accessing menu commands easier, especially when you need to use the same command repeatedly. To tear off a menu, click the dashed line at the top of the menu and drag the menu where you want it. Mesh Tools Normals Color Create UVs E Click and Drag Edit Mesh Connect Components _ | Transform Components ▫▮ Make Hole Tool ᇜ Delete Edge/Nertex Ctrl+Del Merge Edge Tool Detach Multi-Cut Tool Target Weld Tool Paint Reduce Weights Tool

Figure 3.4 Submenus and the all-important option box

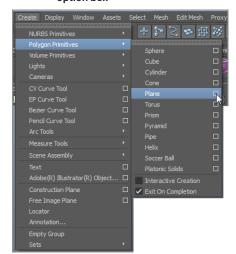
Submenus and the Option Box

You'll notice two different demarcations to the right of some menu items (Figure 3.4): arrows and boxes (called option boxes). Clicking an arrow opens a submenu that contains

> more specific commands. Clicking an option box (□) opens a dialog box in which you can set the options for that particular tool.



Marking menus are a fast UI workflow to allow you to select commands and options without accessing the main menu bar, as you did in the Hypershade in Chapter 2's solar system exercise. For example, right-clicking any object in your scene gives you the marking menu shown in Figure 3.5. With this particular marking menu you can select vertices on that object by moving your mouse to the vertex marking box, as shown in Figure 3.6.



Work Panels and Navigation

The main focus of Maya is its work windows (called *panels*)—the perspective and orthographic views. You use these windows to create, manipulate, and view 3D objects, particles, and animations.

Perspective/Orthographic Panels

The default Maya layout begins with a full-screen perspective view, as shown in Figure 3.7.

By pressing and releasing the spacebar, you can switch your view from the full-screen perspective to the four-panel layout shown in Figure 3.8. Pressing the spacebar again returns your active view panel to Full-Screen mode.

Orthographic views (top, front, and side) are most commonly used for modeling, because they're best at conveying exact dimensions and size relationships.

You can also easily change from perspective to any of the orthographic views in the current panel by clicking the ViewCube () in the upper-right corner of any active panel, though by default the ViewCube is not visible. First you must exit Viewport 2.0 as your viewpanel's renderer by choosing Renderer in the view panel's menu bar, and then switching to Legacy Default Viewport or Legacy High Quality Viewport. Once you are in a legacy view, you can see the ViewCube. To enable/disable the ViewCube when in the legacy viewpanel renderer, in the main menu bar, select Display → Heads Up Display → ViewCube.

SHORTCUTS TO VIEWING

Here's a summary of the most important keyboard shortcuts. Keep in mind that the Option key is used on a Macintosh in place of the Alt key on a PC. See Chapter 2 for more details.

Alt+MMB+Click Tracks around a window.

Alt+RMB+Click Dollies into or out of a view.

Scroll Wheel Dollies into or out of a view.

Alt+LMB+Click Rotates or orbits the camera around in a Perspective window.

Alt+Ctrl+Click and Drag Dollies your view into the screen area specified in your mouse drag.

ViewCube Allows you to change views in a panel easily. This only works in legacy viewports and not the new Viewport 2.0 display that is now default with Maya 2015. In other words, the ViewCube will not display by default, and so the ViewCube will go largely unused, though you may notice it in some figures and images throughout the book.

Macintosh Keys The Option key on a Mac is used as the Alt key on a PC.

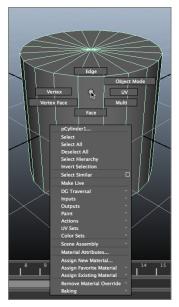


Figure 3.5

A context-sensitive marking menu appears when you right-click an object. Green entries denote new features in Maya 2015.

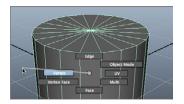


Figure 3.6

By using a marking menu, you can easily select components of an object without using the Status line's icons.

Figure 3.7
The full perspective view

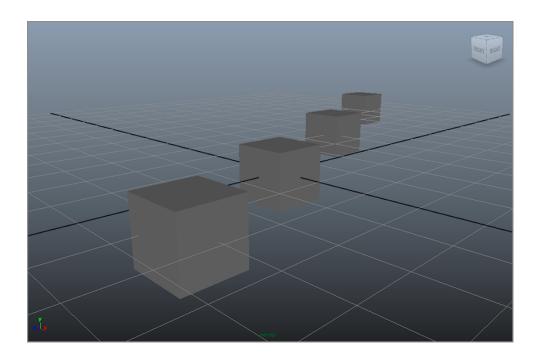
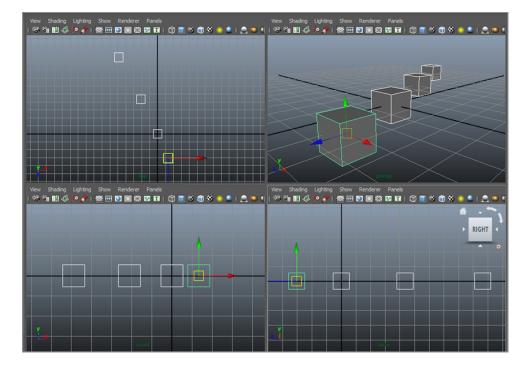


Figure 3.8

The four-panel layout



WIREFRAME AND SHADED MODES

When you're working in the windows, you can view your 3D objects either as wireframe models (as in Figure 3.9) or as solid, hardware-rendered models called Shaded mode (see Figure 3.10). When you press 4 or 5, notice that a text helper opens to tell you your current viewing mode. These messages are called *in-view messages* and can be helpful as you learn the Maya workflow. You can toggle them on or off by choosing, in the main menu bar, Display → Heads Up Display → In-view Messages.

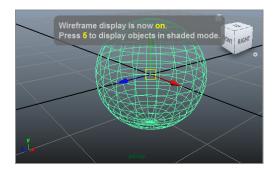


Figure 3.9 Wireframe display of the selected sphere

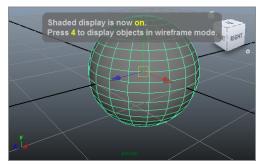


Figure 3.10 Shaded display of the selected sphere

You can cycle through the modes of display by pressing 4, 5, 6, and 7. Wireframe mode is 4, Shaded mode is 5 and shows you a solid view of the objects, Texture Shaded mode is 6 and shows any textures that are applied to the objects, and Lighted mode is 7 and shows a hardware preview of the objects as they're lit in the scene. Table 3.1 gives a quick reference for toggling display levels.

Pressing 5 for Shaded mode lets you see your objects as solid forms and volumes. Pressing 6 for Texture mode is good for the rudimentary alignment of textures. Pressing 7 for Lighted mode (Figure 3.11) is useful for spotting proper lighting direction and object highlights when you first begin lighting a scene.

Other display commands you'll find useful while working in the Modeling windows are found under the panel's View menu. Look At Selection centers on the selected object or objects: Frame All (hotkey = A) moves the view in or out to display all the objects in the scene, and Frame Selection (hotkey = F) centers on and moves the view in or out to fully frame the selected object or objects in the panel.

KEY FUNCTION

Toggles into Wireframe mode

Toggles into Shaded mode

- Toggles into Textured mode
 - Toggles into Lighted mode

Table 3.1 Levels of display detail

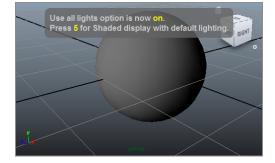


Figure 3.11 Lighted mode (press 7) showing a single light shining on the sphere

The Manipulators

Manipulators are onscreen handles that you use to manipulate the selected object with tools such as Move or Rotate, as you saw in the solar system exercise. Figure 3.12 shows three distinct and common manipulators for all objects in Maya: Move, Rotate, and Scale. In addition, the fourth manipulator shown in Figure 3.12 is the Universal Manipulator, which allows you to move, rotate, or scale an object all within one manipulator.

You can access the manipulators using either the icons from the Tool Box on the left of the UI or the hotkeys shown in Table 3.2.

Table 3.2 Manipulator hotkeys

KEY	FUNCTION
W	Activates the Move tool
E	Activates the Rotate tool
R	Activates the Scale tool
Q	Select tool; also deselects any Translation tools

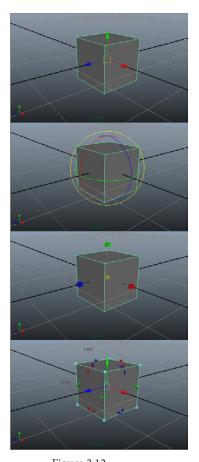


Figure 3.12 Using manipulators

CAPS AND HOTKEYS

Keyboard shortcuts in Maya are case sensitive because in many cases, pressing a single letter key has a different effect than pressing Shift + that letter (which makes the letter uppercase). This book shows all single letters as capitals in the text (the same way they appear on your keyboard). The Shift key is included in the text only when it's part of an uppercase shortcut. So, if you find yourself wondering why pressing a hotkey isn't working, make sure you aren't pressing Shift or that the Caps Lock isn't enabled.

Try This In a new scene, choose Create → NURBS Primitives → Sphere, drag in a view panel on its grid to create a sphere, and then size it however you like. If you have Interactive Creation already turned off for NURBS primitives, a sphere appears at the origin. Press the 5 key in one of the view panels for Shaded mode. In the previous chapter, you tried the manipulators on a sphere to get a feel for how they work. In Chapter 2 you may have noticed the feedback feature on the Universal Manipulator. Select Modify → Transformation Tools → Universal Manipulator, and you'll notice the Universal Manipulator icon appear just below the Tool Box (Manipulate the sphere in the view panel and take a look.

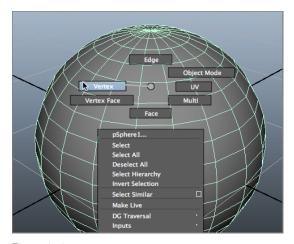
The Universal Manipulator interactively shows you the movement, rotation, or scale as you manipulate the sphere. Notice the coordinates that come up and change as you move the sphere. When you rotate using this manipulator, you see the degree of change. Notice the scale values in dark gray on the three outside edges of the manipulator box; they change when you scale the sphere.

You can scale any manipulator handle. Press the plus key (+) to increase a manipulator's size, and press the minus key (-) to decrease it.

Soft Selection

Soft selection is a way to select part of an object (like a vertex) and manipulate it so that neighboring vertices are affected as well, but in decreasing amounts. Soft selection is best described by seeing it in action.

- 1. In a new scene, create a polygonal sphere at the origin.
- 2. Right-click the sphere to bring up a *marking menu*. Select vertices as shown in Figure 3.13.
- 3. Your display now shows the sphere as cyan and pink points indicating where the vertices are. (I cover vertices in detail in the next chapter.) Manipulating vertices allows you to alter the shape of the polygonal mesh. As you move your mouse over vertices, they turn into red blocks. Click a single vertex to select it.
- 4. Press W for the Move tool and move the vertex away from the sphere, as shown in Figure 3.14. Doing so creates a spike on the sphere.
- 5. This time, let's go into the options for the Move tool. Select Modify → Transformation Tools → Move Tool □ to open the Tool Settings window, shown in Figure 3.15. If you need to expand the window as shown in Figure 3.15, click and drag the vertical bar on the right of the Tool Settings window.
- 6. Scroll down to the Soft Selection heading, and toggle on Soft Select, as shown in Figure 3.16.





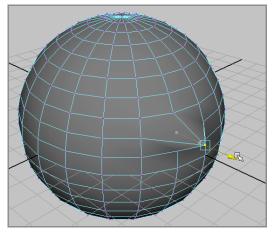
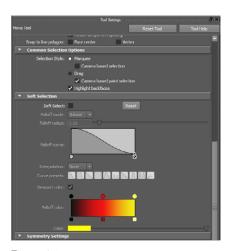


Figure 3.14 Pull a vertex to make a spike.





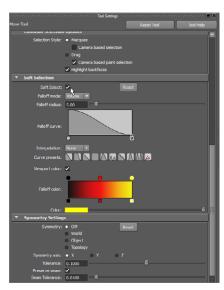


Figure 3.16 Click Soft Select.

- 7. In the persp view panel, right-click the sphere again, and select Vertex. Select another vertex on the sphere. When you do, a gradient of color from yellow to red to black appears on your model. This gradient shows you the influence of your soft selection. Figure 3.17 shows you the falloff region on the sphere.
- 8. Without adjusting anything, move that vertex away from the sphere. This time, instead of a spike forming on the sphere, a much larger, but smooth, bulb forms out of the sphere, as shown in Figure 3.18, much like what would happen if you used the Soft Modification tool.

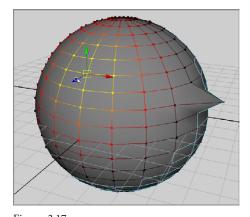


Figure 3.17 Soft Select shows you the falloff gradient.

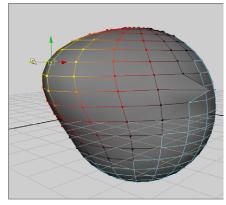


Figure 3.18 Use soft selection to pull out a bulb rather than a spike.

9. You can adjust the size of the falloff area by adjusting the Falloff Radius attribute in the Tool Settings. Be sure to turn off Soft Select and close the Tool Settings window.

Using soft selection on a transform tool such as Move allows you to make organic changes to your mesh easily.

Symmetry

Using the Symmetry transformation option makes symmetrical edits to a mesh. Follow these steps to experience Symmetry with the Move tool:

- 1. Create a polygonal sphere at the origin in a new scene.
- 2. Invoke the Move tool options by double-clicking the Move tool icon in the Tool Box on the left of the UI, shown in Figure 3.19. The options will open just as if you selected the option box through the menu.
- 3. At the bottom of the Tool Settings window (a.k.a. option box) in the Symmetry Settings section, as shown previously in Figure 3.16, click the Object option. You can also set the axis of symmetry to X, Y, or Z. Select Z for now.

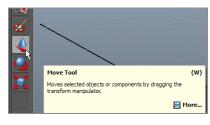


Figure 3.19 Double-clicking the icon will open the options for that tool.

- 4. Enter component selection mode by right-clicking the sphere and choosing Faces from the marking menu that appears.
- 5. Move your cursor over a face and it turns blue, and a face on the opposite side of the sphere turns blue as well. Select a face on one side of the sphere, and a face on the opposite side of the sphere is also selected. Now when you try to move that selected face, the opposite face moves as well in the opposite direction, as shown in Figure 3.20.
- 6. Make sure to turn off Symmetry when you're done.

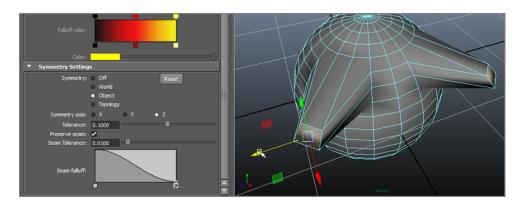


Figure 3.20

The user selected the face on the left and moved it. With Symmetry turned on, the opposite face along the Z-Axis is also selected and moved.

Building a Decorative Box

Let's get back to making things and explore the interface as we go along. In this exercise, you'll build a decorative box, shown in Figure 3.21. You will learn to use reference images for modeling, model polygons with Bevel and Extrude tools, and then add edges with Edge Loop and Interactive Split. Through the process, the Layer Editor helps you stay organized, and you can hide objects from view. This box will be a fairly simple model to make, but you'll use it extensively in Chapters 7, 10, and 11 when working with texture, light, and rendering.

Figure 3.21 A photo of the decorative box



Notice that the box has intricately carved grooves and surface features. You'll build the box to fit the reference and then rely on texture maps created in Chapter 7, "Autodesk" Maya[®] Shading and Texturing" to create the details on the surface of the box. You'll begin by creating *reference planes* in the next section.

Creating Reference Planes

You can use image references from photos or drawings to model your objects in Maya quite easily. These references are basically photos or drawings of your intended model, usually from three different image views of the model (front, side, and top).

The image reference views of the decorative box have already been created and proportioned properly. (You will see a more thorough review of this process for an exercise

Table 3.3

Figure 3.22 Make sure

is toggled off.

Reference views and image sizes

Interactive Creation

in Chapter 6, "Practical Experience.") You can find the images for the box in the Sourceimages folder of the Decorative_Box project. Table 3.3 lists their names, along with their statistics.

FILENAME	VIEW	IMAGE SIZE	ASPECT RATIO
boxFrontRef.tif	Front	1749 x 2023	0.865:1
boxSideRef.tif	Side	1862 x 2046	0.910:1
boxTopRef.tif	Тор	1782 x 1791	1.005:1

The idea here is to map these photos to planes created in Maya. First, press Ctrl+A to toggle off the Attribute Editor if it currently appears to the right of the UI. Toggling off the Attribute Editor displays the Channel Box. Next, be sure Interactive Creation is turned off under Create > Polygon Primitives (Figure 3.22), and then create the reference planes in steps 1 through 3 with the ratios shown in Table 3.4.

1. In the front view panel, create a polygonal plane by choosing Create → Polygon Primitives \rightarrow Plane \square . This plane is for the front image, so in the option box, set Axis to Z, Width to 0.865, and Height to 1.0. Make sure the check box for Preserve Aspect Ratio is deselected, as shown in Figure 3.23. Setting Axis to Z will place the plane properly in the front view. Click Apply to create the plane and keep the option box open.



REFERENCE PLANE WIDTH HEIGHT Front 0.865 1 Side 0.910 1 1.005 1 Top

Table 3.4 Reference planes and sizes

- 2. Switch to the side view panel. Create a second plane, this time with a width of **0.910** and height of 1. Set Axis to X, and make sure Preserve Aspect Ratio is unchecked. Click Apply to create the plane.
- 3. Switch to the top view panel. Create a third plane with a width of 1.005 and height of 1, and set Axis to Y. Make sure the Preserve Aspect Ratio box is still unchecked, and click Create to create the plane and close the option box. Your planes should look like those shown in Figure 3.24.



Figure 3.23 Option box for creating a plane for the front view

- 4. Select the front image plane. In the Channel Box, double-click pPlane1, and rename it to frontPlane. Select the side plane, and rename it from pPlane2 to sidePlane. Rename the top plane from pPlane3 to **topPlane**.
- 5. You still need to place and scale these planes to align them. Take a look at Figure 3.25 to size your reference planes and place them as shown. There are two ways to position these planes. You can manually scale and move them to visually match what you see in Figure 3.25, or you can enter the exact values for scale and translation as shown in Table 3.5 using the Channel Box or Attribute Editor (I discuss these windows next before continuing with the box exercise).

Table 3.5 Reference planes: scale and position

REFERENCE PLANE	XYZ SCALE	XYZ POSITION	
Front	4.711, 4.711, 4.711	0.134, 0.017, -2.167	
Side	4.856, 4.856, 4.856	-1.979, 0, 0	
Тор	4.28, 4.28, 4.28	0, 0, 0.133	

6. Save your work using the main menu bar by choosing File → Save Scene As. Name your work, remembering to use version numbers to keep track of your progress.

You can compare your progress to boxModel01.mb in the Scenes folder of the Decorative_Box project on the book's web page, www.sybex.com/go/introducingmaya2015.

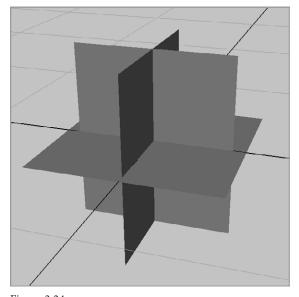


Figure 3.24 The three view planes are ready and waiting at the origin.

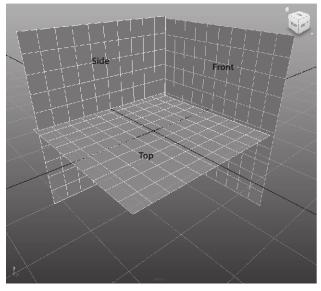


Figure 3.25 Arrange the reference planes for the box model.

The Channel Box/Attribute Editor Explained

To the right of the panels is the Attribute Editor/Channel Box. This is where you'll find (and edit) most of the information, or attributes, about a selected object. Pressing Ctrl+A toggles between the Attribute Editor and the Channel Box.

The Channel Box lists an object's channels that is, the attributes of an object that are most commonly animated. When an object is selected in one of the main views, its name appears at the top of the Channel Box, and its channels are listed. You can edit all the channel values and rename the object itself here.

Toggle on the Attribute Editor by pressing Ctrl+A. This window gives you access to all of a selected object's attributes, whereas the Channel Box displays only the most commonly animated attributes.

Tabs running across the top of the Attribute Editor give you access to the other nodes related to that object, as shown in Figure 3.26.

You can click and drag the top of the Attribute Editor to undock it from the main UI. Once you have it in its own window, pressing Ctrl+A will open the Attribute Editor in its own window from then on. However,

Rotate 0.000

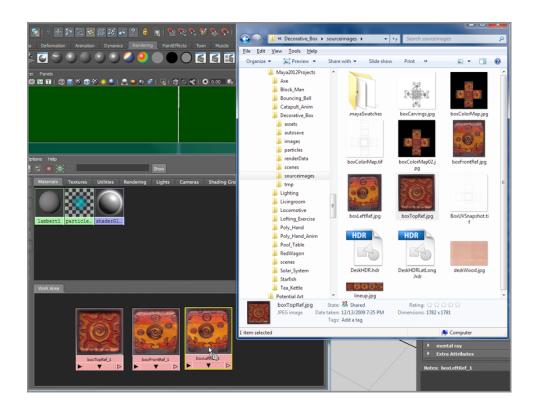
Figure 3.26 The Attribute Editor docked to the main UI

you can dock the Attribute Editor to the main UI by dragging it back over to the Channel Box area. After that, pressing Ctrl+A will toggle between the Channel Box and Attribute Editor again.

Mapping the Box's Reference Planes with Hypershade

Now you'll import the three reference JPEG images from the Sourceimages folder into Maya through the Hypershade window. Click Window → Rendering Editors → Hypershade to open this highly powerful texturing window. In a file browser (Windows Explorer in Windows or the Finder in Mac OS X) window, navigate to the Sourceimages folder of the Decorative_Box project from the companion web page. One by one, select boxFrontRef.tif, boxLeftRef.tif, and boxTopRef.tif and drag them individually into the bottom Work Area section of the Hypershade window, as shown in Figure 3.27.

Figure 3.27 Drag the JPEGs one by one into the Hypershade window.



Once you have imported the JPEG images, the Hypershade displays them in the Work Area. The Hypershade window has tabs along the top. Click the Textures tab, and you will see the three JPEGs there as well. Return to the Materials tab to display your scene's materials, a.k.a. shaders. The bottom Work Area is just that: a work area for you to create and edit materials for your scene. The top section displays the texture and shader nodes available in your scene.

Now you need to create three new shaders to assign to the reference plane objects. You can load the scene file to boxModel01.mb in the Scenes folder of the Decorative_Box project from the companion web page or continue with your own scene.

1. Create three Lambert shaders in the Hypershade by clicking three individual times on the Lambert button in the Create Bar on the left side of the Hypershade window, as shown in Figure 3.28. Simply move the thumbnail icons around in the Work Area so you can see them all. You use the familiar Maya Alt+ LMB, RMB, or MMB key combinations to pan and zoom in the Hypershade window.



Figure 3.28 Create three new Lambert materials.

2. Select the first Lambert node, and press Ctrl+A to open the Attribute Editor. MMB+drag the top reference JPEG thumbnail icon (a.k.a. node) to the Attribute Editor and drop it on the Color attribute for that selected Lambert. Doing so will connect the JPEG to the color of the Lambert material (see Figure 3.29). Rename that Lambert material to topImage.

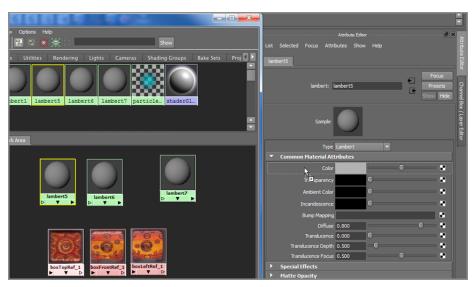


Figure 3.29

Connect the top image JPEG node to the Color attribute of the first Lambert.

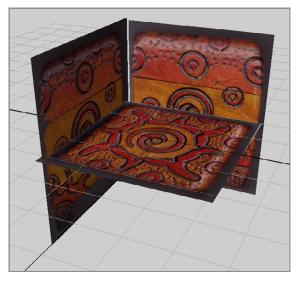


Figure 3.30 The images are applied.

- 3. Repeat step 2 two times to connect and rename the materials for the front and side views. Make sure you label the materials properly; it's tough to tell the side and front images apart.
- 4. To assign the materials, simply MMB+drag each shader's thumbnail icon (or node) from the Hypershade to its respective reference plane in the persp view panel. Press 6 for Texture mode in the Perspective window, as shown in Figure 3.30. Close the Hypershade.
- 5. Choose File → Save Scene As to save a new version of your work.

You can compare your progress to boxModel02.mb in the Scenes folder of the Decorative_Box project at the companion web page.

The Hypershade Explained

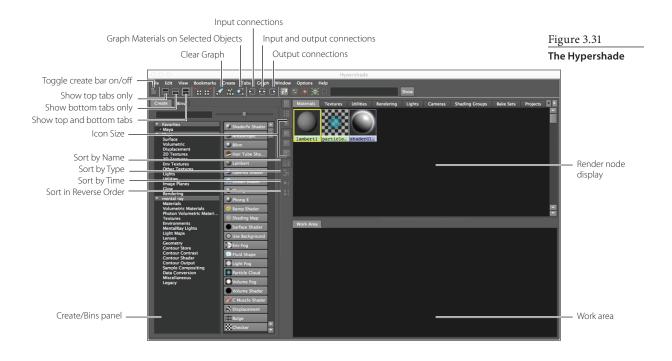
Just as the Outliner window lists the objects in your scene, the Hypershade window lists the textures and shaders. Shaders are assigned to objects to give them their visual appearance—their look and feel.

The Hypershade (Window → Rendering Editors → Hypershade) displays shaders and textures in a graphical flowchart layout (see Figure 3.31). The Hypershade window has three main areas: the Create/Bins panel, the render node display, and the Work Area. The three icons at the upper right let you easily switch views.

The Create/Bins Panel The Create/Bins panel is divided into two tabs, Create and Bins, as shown in Figure 3.31 on the left side of the window. Selecting the Create tab gives you access to creating a variety of render nodes (shaders and textures, for example) by clicking them. The Bins tab adds a level of organization by letting you store sets of shaders in different bins to sort them. The bar at the top switches between Create Maya Nodes and Create Mental Ray Nodes.

The Render Node Display Area After you create a render node, it appears in the display area as a thumbnail icon as well as in the Work Area and is available for editing. Doubleclicking an icon opens the Attribute Editor. You can also use the MMB to drag any icon to the Work Area, where you can work on making or editing shading networks.

The Work Area The Work Area is a free-form workspace where you can connect render nodes to form-shading networks that you can assign to your object(s) for rendering. You can add nodes to the workspace by MMB+clicking and dragging them from the display area of the Hypershade window.



Organizing Workflow with the Layer Editor

Now that you have the reference planes set up and mapped, you'll create *display layers* to help organize the scene before you actually start modeling. You can load the scene file

boxModel02.mb in the Scenes folder of the Decorative Box project from the companion web page or continue with your own scene.

- 1. Select the three reference planes, and toggle off the Attribute Editor to show the Channel Box. Under the Channel Box is the Layer Editor. Click the Display tab, as shown in Figure 3.32.
- 2. With the planes selected still, click the Create A New Layer And Assign Selected Objects icon at the top of the Layer Editor, as shown in Figure 3.33. Doing so creates a new layer for these three reference planes.
- 3. In the Layer Editor, double-click the name layer1 to open the Edit Layer window. Rename the layer to referenceLayer2, as shown in Figure 3.34.





Figure 3.32 The Display tab in the Layer Editor

Figure 3.33

Click to create a new display layer and add the selected objects automatically.



Figure 3.34 Name the new display layer.



Figure 3.35 Toggle the visibility of the reference layer.

- 4. Toggle the display of this layer by toggling the V icon, shown in Figure 3.35.
- 5. Save another version. You can compare your progress to box-Model03.mb in the Scenes folder of the Decorative_Box project.

Display layers allow you to easily turn on and off the display of the reference planes as you model the decorative box. Become familiar

with this feature early because it will be a valuable asset when you animate complicated scenes.

To add items to an already created layer, select the objects; then right-click the desired layer and choose Add Selected Objects. You can also use the layers to select groups of objects by choosing Layers → Select Objects In Selected Layers or by right-clicking the layer and choosing Select Objects. To change the name and color of a layer, double-click the layer to open the Edit Layer window, as shown earlier in Figure 3.34.

Modeling the Decorative Box

Make sure you are in Texture mode (press 6) so you can see the reference plane and the images on them in the persp view panel. Also be sure to toggle on visibility of the references' display layer. In Chapter 4, I'll cover in more detail the modeling tools you'll use.

You can load the scene file boxModel03.mb in the Scenes folder of the Decorative_Box project from the companion web page or continue with your own scene. To model the box to fit the references, follow these steps:

- 1. With Interactive Creation turned off, select Create → Polygonal Primitives → Cube. Position and size the cube to roughly match the size of the reference planes for the decorative box.
- 2. To make it easier to see the reference planes and their images in relation to the box you just created, in the menu bar of the persp view panel, select Shading → X-Ray, as shown in Figure 3.36. Now you can see the box and the reference images at the same time.
- 3. Scale and position the cube to match the size of the main part of the box, as shown in Figure 3.37. Don't bother sizing the box to include the little feet on the bottom of the box. Use X-Ray mode in the Side, Front, and Top modeling panels in Maya to line up the cube as best as you can. This will be the base model for the decorative box.
- 4. Switch off X-Ray mode in your views (in the persp view panel's menu bar, select Shading → X-Ray). Let's work on the rounded bevel on top of the box, where the lid is. Select the poly cube, open the Attribute Editor (Ctrl+A), and click the pCubeShape1 tab to access the shape node attributes.

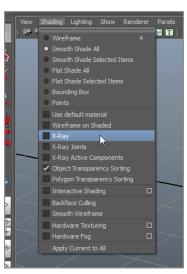


Figure 3.36 Set the display to X-Ray mode so you can see how the poly cube and the decorative box line up.

- 5. Under Object Display → Drawing Overrides, check Enable Overrides to enable it. Deselect Shading to display the poly cube as a wireframe while the reference planes remain displayed as textured planes. This way, you can more easily match the cube to the decorative box (see Figure 3.38).
- 6. Right-click the box and select Edge from the marking menu, as shown in Figure 3.39.
- 7. Select the top four edges of the cube and switch to the front view, as shown in Figure 3.40. Using the front view, you'll shape the top of the cube.
- 8. Make sure you are in the Polygons menu set; then select Edit Mesh → EDGE → Bevel. Don't worry about the settings; you'll adjust them after the fact. You should now have something like Figure 3.41 if your bevel options were at the defaults. Because you created a bevel operation on the cube, Maya has created a new node connected to the cube. You will access this bevel node to adjust the bevel settings on the cube.

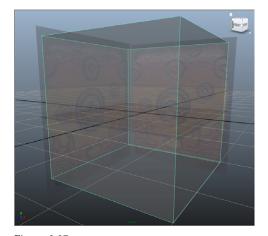


Figure 3.37 Size the cube to fit the box references.

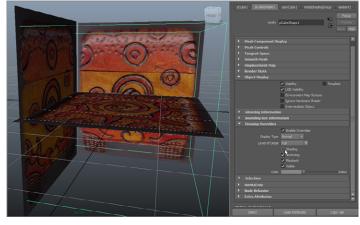


Figure 3.38 Display the cube as a wireframe.



Figure 3.39 Select Edge from the marking menu.



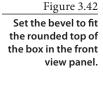
Figure 3.40 Select the top four edges.



Figure 3.41 Default bevel

- 9. Right-click the cube, select Object Mode from the marking menu, and then select the cube. Toggle on the Attribute Editor, and select the new polyBevel1 tab. Using the front view panel, set Offset so that it lines up with the rounded top of the box, at about **0.26**. Set Segments to **12**. Make sure Auto Fit, Offset As Fraction, and World Space are all checked. Also make sure UV Assignment is set to Planar Project Per Face (see Figure 3.42).
- 10. In the side view panel, move the bottom corner vertices on the cube to line up the bottom corners of the box to the reference image.
- 11. In the front view panel, move the bottom corner vertices to match the bottom of the box in the image (see Figure 3.43). Don't worry about the curvature in the middle of the box or the feet just yet. Save your work.
- 12. Let's take a quick look at how this box will look rendered now. Click anywhere in the persp view panel to make it the active panel. In the Status line (the group of icons at the top of the screen), click the Render The Current Frame icon, as shown in Figure 3.44.
- 13. Save your work. You can compare your progress to boxModel04.mb in the Scenes folder of the Decorative_Box project from the companion web page.

When you rendered your work in step 12, the Render view opened to show you a gray shaded box with the reference planes barely showing, as you can see in Figure 3.45.



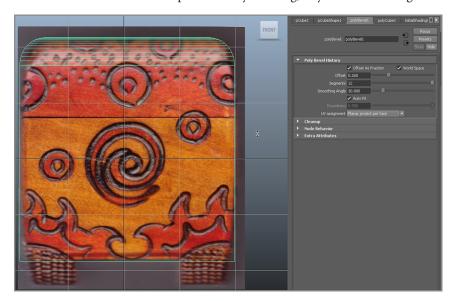
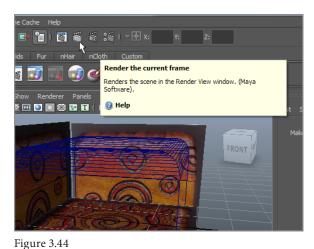




Figure 3.43 Taper the bottom of the cube.

Figure 3.46





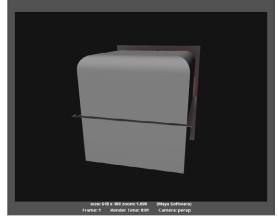
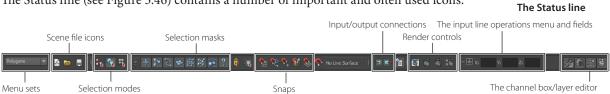


Figure 3.45 The model thus far is rendered.

Status Line Explained

The Status line (see Figure 3.46) contains a number of important and often used icons.



The Status line begins with a drop-down menu that gives you access to the menu sets in Maya. You'll notice that intermittently throughout the Status line are white vertical line breaks with either a box or an arrow in the middle. Clicking a break opens or closes sections of the Status line.

Some of the most often used icons are identified here.

Scene File Icons

The tools in the first section of the Status line deal with file operations; they are Start A New Scene (), Open An Existing Scene (), or Save Your Current Scene ().

Selection Modes

Selection modes allow you to select different levels of an object's hierarchy (see Table 3.6). For example, using a selection mode, you can select an entire group of objects, only one of the objects in that group, or even components (vertices, faces, and so on) on the surface of that object, depending on the selection mode you're in.

Table 3.6
Selection modes

ICON	NAME	DESCRIPTION
- T ₃	Hierarchy and Combinations mode	Lets you select groups of objects
	Object mode	Lets you select objects such as geometry, cameras, lights, and so on
-	Component mode	Lets you select an object's components, such as vertices, faces, UVs, and so on

To switch between the two most often used modes—Object and Component—press the F8 key. You can also select among the Component and Object modes from the marking menu when you right-click an object.

You'll work with these selection mask filters throughout the book, but you will likely access them through marking menus as you have already done to select vertices and edges of a polygonal object. For a quick preview, hover your cursor over each of the icons to see a tooltip that gives the icon's name and describes its function.

Snapping Functions, or Snaps

The icons with the magnets are called *snaps*. They allow you to snap your cursor or object to specific points in the scene, as you saw in the solar system exercise. You can snap to other objects, to CVs or vertices (), and to grid intersections () and other locations by toggling these icons. Table 3.7 shows the various snaps.

Figure 3.47 Attribute Editor/ Channel Box/Tool Settings icons



The Channel Box/Layer Editor Icons

These last four buttons on the Status line (Figure 3.47) toggle between the Attribute Editor, Channel Box, and Modeling Toolkit view on the right side of the UI. Clicking the first icon (🔡) shows or toggles the Modeling Toolkit. The second icon (🗐) toggles the Attribute Editor, much the same as pressing Ctrl+A. The third icon (I) displays or hides the Tool Settings window along the left side of the UI, as you've seen with soft selections. The fourth icon here (🐸) toggles the display of the Channel Box, again much the same as pressing Ctrl+A.

ICON	NAME	DESCRIPTION	Table 3.7
<u>•</u>	Snap To Points	Lets you snap objects to object points such as CVs or vertices.	Snap icons
	Snap To Grids	Lets you snap objects to intersections of the view's grid.	
S	Snap To Curves	Lets you snap objects along a curve.	
	Snap to Projected Center	Lets you snap to the center of a selected object.	
	Snap To View Planes	Lets you snap objects to view planes.	
○	Make The Selected Object Live	This icon has nothing to do with snapping but is grouped with the Snap To icons. It lets you create objects such as curves directly on a surface.	

Editing the Decorative Box Model Using the Shelf

Back to work on the box model. You will use the Shelf in the UI to access some of the commands for the next series of steps as you continue working on the box. The Shelf runs directly under the Status line and contains an assortment of tools and commands in separate tabs, as shown in Figure 3.48.



Figure 3.48 The Shelf

You can load the scene file boxModel04.mb in the Scenes folder of the Decorative_Box project from the companion web page or continue with your own scene.

In the following steps, you have to add surface detail to the model so you can more adequately adjust its shape:

1. Orient the persp view panel so you can see the bottom of the box and then select the box model. In the Shelf, click the Polygons tab. Double-click the Interactive Split Tool icon to open its Tool Settings, as shown in Figure 3.49. Set Snap Magnets to 0. The Interactive Split tool is only selectable through the Shelf and is no longer shown in any of the menus.

2. Your cursor will change to a solid triangle. Click, as shown in Figure 3.50, to select a point along the bottom-left edge of the box that corresponds with where the box's feet begin.

3. Click a second point on the opposite edge on the bottom of the box to create a new edge line, as shown in Figure 3.51. Click the RMB to commit the new edge line. This creates surface detail along the bottom of the box for you to model the feet for the box. This methodology is explained in detail in Chapter 4.



The Interactive Split Tool icon in the Shelf









Figure 3.51 Create a new edge line along the bottom of the box.

The Interactive Split Tool you have just used in Steps 1-4 of this exercise is being phased out of the Maya interface starting with Maya 2015. You can currently only find this still-popular tool in the Shelf. It's functionality is being replaced by the Multi-Cut tool found in the Modeling Toolkit, which we will be exploring in the next chapter.

- 4. Using the same procedures in steps 1 through 3, create three more edge lines for a total of four separate splits in the bottom face of the cube that line up with the legs of the box, as shown in Figure 3.52. The preceding three steps have created a surface detail called *faces* that allow you to create the feet for the box.
- 5. Click the Select Tool icon in the Tool Box to the left of the UI (shown in Figure 3.53) to exit the Interactive Split tool. Your cursor returns to the regular Maya cursor.



Figure 3.52 Split the bottom face four times to create divisions for the box's feet.

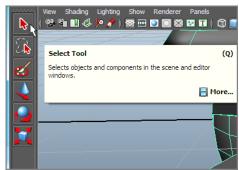


Figure 3.53 The Select tool in the Tool Box

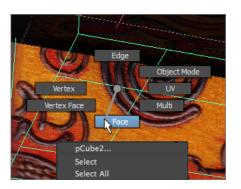


Figure 3.54 Select Face from the marking menu.

6. Right-click the box, and select Face from the marking menu (Figure 3.54). As you hover your mouse over the parts of the box, the new faces in the four corners of the box you just created will highlight in red. Shift+click the four corner faces to select all four faces, as shown in Figure 3.55.

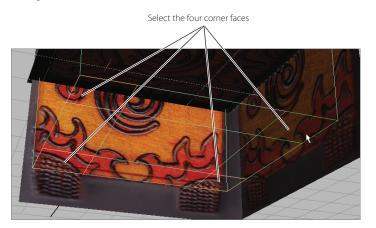


Figure 3.55 Select the four corner faces.

7. With the four faces selected, go into the Shelf and select the Extrude icon shown in Figure 3.56. This tool is also accessible through the menu Edit Mesh → FACE → Extrude.



Figure 3.56 The Extrude icon in

the Shelf

- 8. Your manipulator will change, as shown in Figure 3.57. Grab the Z-axis move handle, and drag it down to "pull" the feet out of the bottom of the box as shown.
- 9. By moving vertices, taper the feet to match the reference images in the front and side view panels; see Figure 3.58.
- 10. Save your work. You can compare your progress to boxModel05.mb in the Scenes folder of the Decorative_Box project.

Figure 3.57 Extrude the feet.

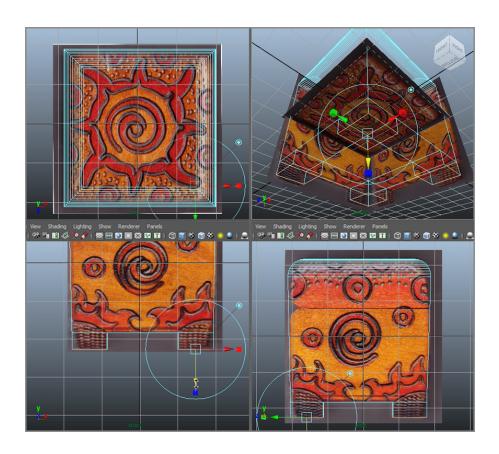
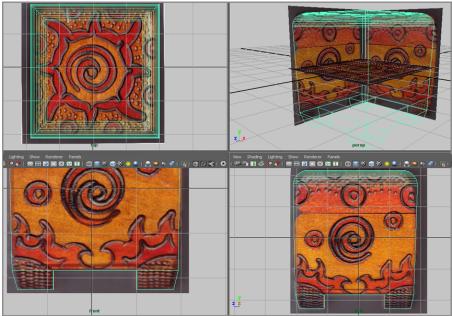


Figure 3.58 Move the vertices on the feet to line them up to the reference images.



The Shelf and Tool Box Explained

Here is a brief explanation of the tools and icons in the Shelf and Tool Box.

The Shelf

The Shelf, shown earlier in Figure 3.48, is an area where you keep icons for tools. It's divided into tabs that define functions for the tool icons in the Shelf. Don't worry too much about the Shelf right now; it may be better to use the commands from the menus first before turning to icons and shelves.

The Tool Box

I C O N

NAME

The Tool Box, shown in Figure 3.59, displays the most commonly used tools. Table 3.8 lists the icons and their functions.

In addition to the common commands, the Tool Box displays several choices for screen layouts that let you change the interface with a single click. Experiment with the layouts by clicking any of the six presets in the Tool Box.

DESCRIPTION



Figure 3.59 The Tool Box

Table 3.8 **Tool Box icons**

ICON	NAME	DESCRIPTION
	Select	Lets you select objects
[Ch	Lasso Select	Allows for a free-form selection using a lasso marquee
	Paint Selection Tool	Enables the Paint Selection tool
	Translate (Move)	Moves the selection
	Rotate	Rotates the selection
	Scale	Scales the selection
	Last Tool Used	Shows the last tool that was used (shown as Split Polygon Tool here, sometimes shown blank)

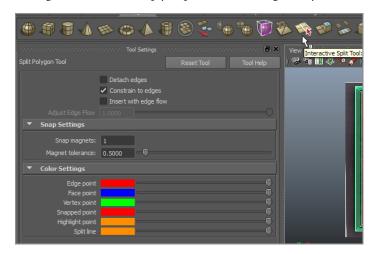
Continuing the Decorative Box Model

Back to work! You'll be spending more time getting the box in shape. You can load the scene file boxModel05.mb in the Scenes folder of the Decorative_Box project or continue with your own scene. In the following steps, you will add more faces and edges to the model surface (a.k.a. mesh) so you can add detail to the shape:

1. The middle of the box has a bit of a curve; you will need to create a new edge line that runs across the middle of the box so you can bow out the sides in the middle. In the Layer Editor, toggle off visibility for the reference images by clicking the wicon for reference planes. Now you can see just the model of the box.

2. Select the box, and in the Polygons tab of the Shelf, double-click the Interactive Split Tool icon () for the options and set Snap Magnets to 1 and the Magnet Tolerance to 0.5, as shown in Figure 3.60. (This tool is only selectable through the Shelf, and not through the main menu bar as of Maya 2015.) You will be using the Snap Magnets setting for this tool to help you place the new edge line you're about to create.

Figure 3.60 Interactive Split tool options



- 3. Move your triangle cursor along one of the box's edges. With Snap Magnets set to 1, you will notice the cursor will snap at the corners and in the middle of the edge. Drag the mouse along the box's first edge to snap the placement in the middle of that edge, as shown in Figure 3.61.
- 4. Snap a new split point in the middle of the next edge of the box, creating an edge line along one side of the box. Continue to snap two more split points in the middle of each remaining edge to create a horizontal split line in the middle of the box, as shown in Figure 3.62. When you place your last point at the same point where you started, right-click to commit the action. You now have a horizontal split along the middle of the box.
- 5. Click the Select Tool icon in the Tool Box to the left of the UI () to exit the Interactive Split tool and close the Tools Settings panel to gain more screen space. Your cursor returns to the regular Maya cursor. Turn on the referencePlanes layer to show the image references.
- 6. Right-click the box, and choose Vertex from the marking menu. Move the new vertices in the middle of the box to bow out the box slightly. Move the rest of the vertices to match the model to the box images in the side and front view panels (see Figure 3.63).
- 7. Save your work. You can compare your progress to boxModel06.mb in the Scenes folder of the Decorative_Box project.

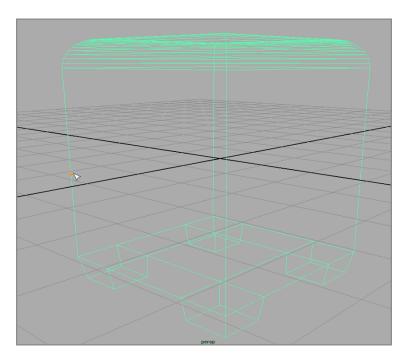


Figure 3.61

Click and drag the Interactive Split tool on this edge of the box and snap the point at the middle.

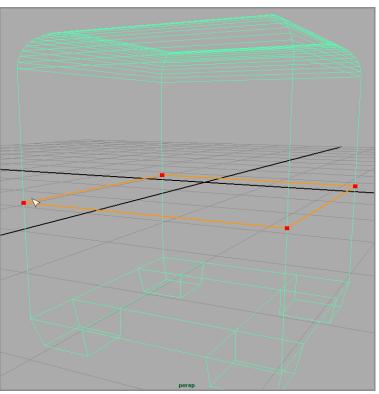


Figure 3.62 Create a horizontal edge line all the way

around the box.

Figure 3.63 Adjust the cube to fit the reference images.



Time Slider and Help Line Explained

In this section, you will examine the bottom part of the UI where the Help line and Time slider live.

Time Slider/Range Slider

Running horizontally across the bottom of the screen are the Time slider and the Range slider, as shown in Figure 3.64. The Time slider displays the range of frames available in your animation and gives you a gray bar, known as the Current Time indicator. You can click it and then drag it back and forth in a scrubbing motion to move through time in your sequence. (When instructed in this book to scrub to a certain point in your animation, use this indicator to do so.)

Figure 3.64 The Time and Range sliders



The text box to the right of the Time slider gives you your current frame, but you can also use the text box to enter the frame you want to access. Immediately next to the current time readout is a set of DVD/DVR-type playback controls that you can use to play back your animation.

Figure 3.65

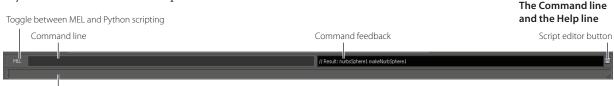
Below the Time slider is the Range slider, which you use to adjust the range of animation playback for your Time slider. The text boxes on either side of this slider give you readouts for the start and end frames of the scene and of the range selected.

You can adjust any of these settings by typing in these text boxes or by lengthening or shortening the slider with the handles on either end of the bar. When you change the range, you change only the viewable frame range of the scene; you don't adjust any of the animation. Adjusting the Range Slider lets you zoom into sections of the timeline, which makes adjusting keyframes and timing much easier, especially in long animations.

Command Line/Help Line

Help line

Maya Embedded Language (MEL) is the user-accessible programming language of Maya. Use the Command line (see Figure 3.65) to enter single MEL commands directly from the keyboard in the white text box portion of the bar.



Below the Command line is the Help line. This bar provides a quick reference for almost everything on the screen. It also prompts you for the next step in a particular function or the next required input for a task's completion. The Help line is useful when you're not really sure about the next step in a command, such as which object to select next. You'll be surprised by how much you'll learn about tool functions by reading the prompts displayed here.

Finishing the Decorative Box Model

Now that you have the overall shape of the box finished, you need to add a few finishing details to the box. You will round out the edges of the box so they are not sharp, as well as add a line around the top of the box for the lid's seam and hinges. You can load the scene file boxModel06.mb in the Scenes folder of the Decorative_Box project or continue with your own scene.

To make a model more dynamic, you can round or bevel the edges to heighten the realism of the model when it is lit and rendered.

1. Let's turn off the image reference planes again, but this time, let's do it a different way. Open the Outliner window by choosing Window → Outliner. You are already somewhat familiar with the Outliner from the solar system exercise in the previous chapter. Select frontPlane, sidePlane, and topPlane in the Outliner, and hide them by

Figure 3.66 Hiding objects using the Outliner



- selecting Display → Hide → Hide Selection or by pressing Ctrl+H. This way, you can individually hide any object in your scene. Notice that when an object is hidden, its Outliner entry is grayed out (Figure 3.66).
- 2. Select the box, and open the Attribute Editor. Under the Object Display → Drawing Overrides headings, uncheck the Enable Overrides box, as shown in Figure 3.67, to display the box in Shaded mode again. This will help you see how the bevel works.
- 3. You will bevel the edges of the box throughout to soften the crisp corners of the cube that the real box doesn't have. Right-click the box and choose Edge from the marking menu. Shift+select all the outer edges of the cube, as shown in Figure 3.68.
- 4. With those edges selected, select Edit Mesh → EDGE → Bevel □. Set everything to the defaults, but change Segments from 1 to 3. Click Bevel, and your box should resemble the one shown in Figure 3.69.
- 5. Select the cube, and delete its history by selecting Edit → Delete By Type → History. This process essentially cleans up the model and the procedures it has undergone. You'll learn about History in the following chapters.
- 6. You have one final detail to tend to on the lid. You need to add the hinge area you can see in the side view panel's reference image. Select the reference planes in the Outliner (pPlane1, pPlane2, and pPlane3), and press Shift+H to unhide them (you can also choose Display → Show → Show Selection). Make sure you're in Texture Shaded mode in your views (press 6) to see the box images.

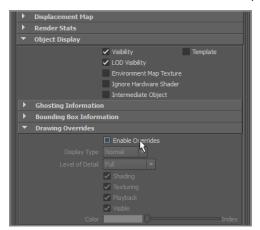
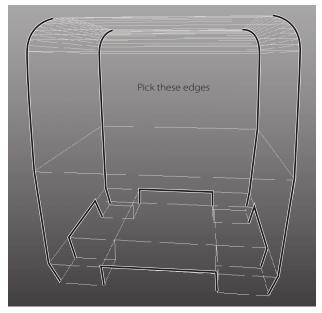


Figure 3.67 **Turn off Drawing** Overrides in the

Attribute Editor to display the box in Shaded mode again.

- 7. Select the box. Turn its display back to wireframe by going into the Attribute Editor and, in the pCubeShape1 tab, turning on Enable Overrides under the Object Display → Drawing Overrides heading.
- 8. Select Mesh Tools → EDIT → Insert Edge Loop Tool. This tool is like the Interactive Split tool in that it inserts new edges into a model. Your cursor will change to the solid triangle again. Click the upper edge of the box and drag a dashed line to line up with the lid's seam in the box reference images, as shown in Figure 3.70. Once placed and you release the mouse button, the edge line will be completed, and the dashed line will turn solid.



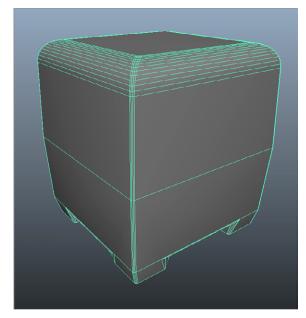


Figure 3.68 Select all of these edges for beveling.

Figure 3.69 The beveled edges of the box

9. In the side view panel, insert four more horizontal edge loops for a total of five edge loops, as shown in Figure 3.71. This gives you edges with which to create the wedge cutout where the box is hinged, and that gives you a little indentation where the lid meets the box. You won't create a separate lid because you won't animate the box to open or close, and you don't need to see the inside.



Figure 3.70 Insert an edge loop to line up with the seam in the real box.

Figure 3.71

Insert these five edge loops for the lid of the box.



10. In the side view panel, select the appropriate vertices (see Figure 3.72), and move them to create the wedge-shaped indentation as shown.

Figure 3.72

Move the vertices to create the hinge area in the back of the box.



- 11. Choose Select → Select Edge Loop Tool. Select the middle edge loop you created earlier for the indent where the lid meets the box, as shown in Figure 3.73. Press R to scale the edge loop very slightly inward, as shown.
- 12. Hide the reference planes again through the Layer Editor and turn Shading back on for the cube in the Attribute Editor. Figure 3.74 (left) shows the completed box. But there's still a little snag. Notice the dark area where the lid meets the box, where you just created the slightly indented seam line. This is because of Normals. It makes the lid look as if it's angled inward.



Figure 3.73 Select this edge loop, and scale it to create an indent line where the lid meets the box.

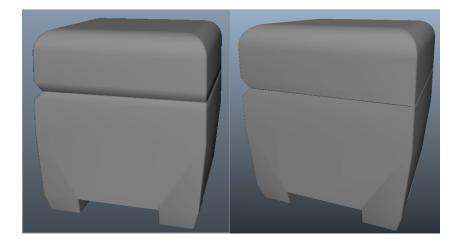


Figure 3.74 The completed box needs one more adjustment (left). The box now looks better after adjust-

ing Normals (right).

- 13. Select the box and choose Normals → Set Normal Angle. In the Set Normal Angle window that pops open, set Angle to the default of 30, and click Apply And Close. Doing so fixes the darkening, as shown in Figure 3.74 (right). For more on Normals, see the "Normals" sidebar. Select the box and delete its history by choosing Edit → Delete By Type → History.
- 14. Save your work, grab someone you love, and give them a hug.

You're finished with the modeling portion of this decorative box, and you've gotten to know the interface much better. In later chapters, you'll texture, light, and render the box with photorealism in mind. You can load boxModel07.mb from the Scenes folder in the Decorative_Box project to compare your work.

NORMALS

Normals are imaginary lines that are perpendicular to a mesh's poly face and that define sides for that face. They also help determine how a renderer, such as mental ray®, shades the surface. In some cases when you're modeling, you may notice an action that causes part of your model to display a darkened area as you saw in the decorative box in Figure 3.74. By manually setting a Normal angle for the box as you did in step 13 of the exercise, you override the display anomaly. You'll learn more about Normals in Chapter 7.

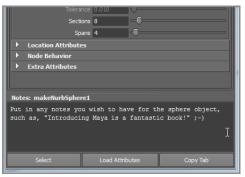
The Attribute Editor and Outliner Explained

You have worked with the Attribute Editor and Outliner several times already. Here's a brief overview of these all-important windows in the workflow in Maya.

The Attribute Editor Window

To use the Attribute Editor, select Window → Attribute Editor (Ctrl+A). The Attribute Editor window is arguably the most important window in Maya. As you've already seen, objects are defined by a series of attributes, and you edit and even set keyframes for these attributes using the Attribute Editor. Some attributes listed in the Attribute Editor are also shown in the Channel Box. These attributes, despite being shown in two places, are the same.

Figure 3.75 You can keep notes with an object's attributes in the Attribute Editor.



The Attribute Editor has tabs that correspond to the object's node structure. You learned a little about the Maya object structure in the previous chapter.

You'll see an area for writing notes at the bottom of the Attribute Editor. This is handy because you can put reminders here of important events, such as how you set up an object or even a birthday or an anniversary. If you drag the horizontal bar, you can adjust the size of the notes space, as shown in Figure 3.75.

The Outliner

The Outliner is perfect for organizing, grouping objects, renaming nodes, and so forth, as you've already seen.

To use the Outliner, select Window → Outliner (see Figure 3.76). It displays all the objects in your scene as an outline. You can select any object in a scene by clicking its name.

The objects are listed by order of creation within the scene, but you can easily reorganize them by MMB+clicking and dragging an object to a new location in the window. This is a fantastic way to keep your scene organized. Additionally, you can easily rename an object by double-clicking its Outliner entry and typing a new name.

A separator bar in the Outliner lets you split the display into two separate outline views. By clicking and dragging this bar up or down, you can see either end of a long list, with both ends having independent scrolling control.

Introducing the Modeling Toolkit

As you've seen in the interface, alongside the right of the UI where the Attribute Editor and Channel Box reside is a third tab called Modeling Toolkit, shown in Figure 3.77. This suite of tools makes polygon modeling more efficient since the most often used tools are centralized into one place. In addition, the Modeling Toolkit, when activated, allows for faster and easier component selection and editing.

In the Modeling Toolkit, the top half centers around making selections, while the bottom half lists important polygon workflow tools such as Bevel and Extrude. All of the Modeling Toolkit tools work slightly differently than the standard Maya tools of the same name; however, the results of the executed tool are identical. This tool set is explored in depth and put to good use in Chapter 4.







Modeling Toolkit tab

Summary

In this chapter, you learned more about the user interface and the primary windows used in Maya as you worked on modeling the decorative box. The user interface combines mouse and keyboard input as well as plenty of menu and tool icons that you can select and use to accomplish your tasks.

You'll be quizzed in 10 minutes. Do you have it all memorized? Don't worry if you haven't absorbed all the information in this chapter. Now that you've had some exposure to the Maya user interface, you'll be familiar with the various windows when you really get to work. You can always come back to this chapter to refresh your memory. Remember, you should learn the Maya program using its default settings. When in doubt, remember to access the Maya Help system (F1 keyboard shortcut or the Help menu in the main menu bar).



Beginning Polygonal Modeling

Simple objects call for simple models, and complicated objects call for a complex arrangement of simple models. Like a sculptor, you must analyze the object and deconstruct its design to learn how to create it.

The Autodesk* Maya* software primarily uses two types of modeling: polygons and NURBS. Both require a process that begins with deciding how best to achieve your design, although it's common to mix modeling methods in a scene.

To help you decide where to begin, this chapter starts with an overview of modeling, briefly describing the two popular methods and how they differ. You'll also learn about primitives. The second part of the chapter takes a detailed look at modeling with polygons. (The next two chapters cover the process of modeling with polygons and NURBS surfaces and how to bring them together in one model.)

Learning Outcomes: In this chapter, you will be able to

- Decide how to plan your model
- Edit polygon geometry in traditional Maya as well as Modeling Toolkit workflows
- Navigate the Modeling Toolkit interface
- Work with the Modeling Toolkit selection workflow
- Extrude, bevel, and wedge polygons
- Use edge loops to create detail
- Create curves and use the Revolve function to convert them to polygon meshes
- Adjust grouping and hierarchies in a complex model

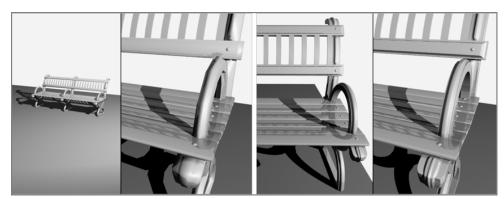
Planning Your Model

When you dissect the components of an object into primitive shapes, you can then translate and re-create the object in 3D terms.

First, you should take reference pictures from many angles, get dimensions, and even write down a description of the object. The more perspectives from which you see your subject, the better you'll understand and be able to interpret your model.

You must also decide the purpose for your model and determine the level of detail at which it will be seen in your CG scene. Consider the two scenes in Figure 4.1. If you need to create a park bench for a far shot (left), it will be a waste of time and effort to model all the details such as the grooves in the armrest. However, if your bench is shown in a closeup (the images on the right), you'll need those details.

Figure 4.1 The level of detail you need to include in a model depends on how it will be seen in the animation.



If you aren't certain how much detail you'll need, it's better to create a higher level of detail rather than skimping. You can more easily pare down detail than create it later.

Keep in mind that you can also add detail to the look of your model in the texturing phase of production, as you'll see with the decorative box later in the book. (Chapter 7, "Autodesk" Maya Shading and Texturing," covers texturing.)

Choosing a Method

Polygon modeling involves tearing and extruding from larger pieces to form a desired shape. This method is typically preferred by most digital artists in the field.

NURBS modeling is great for organic shapes because smooth lines, or *curves*, are the basis of all NURBS surfaces. However, NURBS tends to be more difficult in comparison since it's more difficult to create a complete model without several surfaces that must be perfectly stitched together, a process not covered by this book. Subdivision surfaces combine the best of both worlds but are not a popular workflow and is largely hidden in the interface now; therefore, it will not be covered in this book. Basic NURBS modeling is covered in the next chapter.

An Overview of Polygons

Polygons consist of faces. A single polygon face is a flat surface made when three or more points called *vertices* are connected. The position of each *vertex* defines the shape and size of the face, usually a triangle. The line that connects one vertex to another is called an *edge*. Some polygonal faces have four vertices instead of three, creating a square face called a quad.

Polygonal faces are attached along their polygonal edges to make up a more complex surface that constitutes your model (as shown with the polygonal sphere in Figure 4.2). A camping tent is a perfect example. The intersections of the poles are the faces' vertices. The poles are the edges of the faces, and the cloth draped over the tent's frame is the resultant surface.

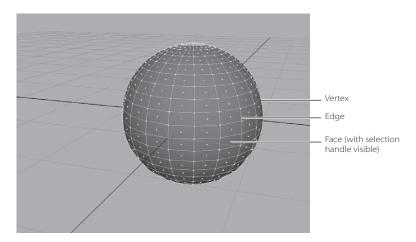


Figure 4.2 A polygonal sphere and its components

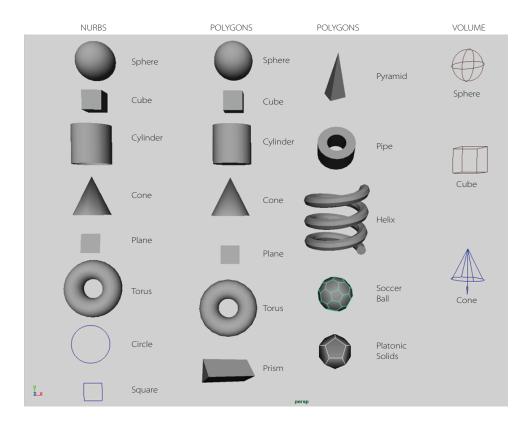
Polygon models are the simplest for a computer to render. They're used for gaming applications, which need to render the models as the game is running. Gaming artists create models with a small number of polygons, called low-count poly models, which a PC or game console can render in real time. Higher-resolution polygon models are frequently used in television and film work.

Using Primitives

Primitives are the simplest objects you can generate in Maya (or in any 3D application). They are simple geometric shapes—polygons or NURBS. Typically, primitives are used to sculpt models because you can define the level of detail of the primitive's surface; they offer great sculpting versatility through vertex manipulation.

To get a better sense of how to begin a modeling assignment, you may find it helpful to analyze your modeling subjects into forms and shapes that fit in with Maya primitives. Figure 4.3 shows all of the primitives in Maya, including NURBS, polygons, and volume primitives. Quite different from geometry primitives, volume primitives are used for lighting and atmosphere effects, such as fog or haze, and don't play a part in modeling.

Figure 4.3
The Maya primitives



Polygon Basics

Polygon modeling is popular because its resulting models are usually one piece of geometry with many facets. You can, therefore, deform polygon models without fear of patches coming apart, as can happen with NURBS. Polygons, however, have a finite detail limitation and can look jagged up close or when scaled up. One solution to this problem in the Maya software is the Smooth tool.

A popular method of polygonal modeling, sometimes called *box modeling*, involves creating a base object, such as a simple cube, and then pulling and pushing faces to draw angles to create more faces. Whereas you typically need to create curves to start NURBS, you usually create complex polygons from basic-shaped polygons such as primitives.

A second method for creating poly surfaces uses the same curves that NURBS surfaces use or even converts a completed NURBS surface model to polygons. A third method is to create poly surfaces directly with the Polygon tool, which allows you to outline the shape of each face.

Creating Polygonal Primitives

With a poly mesh, detail is defined by subdivisions, which are the number of rows and columns of poly faces that run up, down, and across. The more subdivisions, the greater definition and detail the mesh is capable of.

Choosing Create - Polygon Primitives gives you access to the poly version of most of the NURBS primitives. Opening the option box for any of them gives you access to their creation options. To see an example, choose Create → Polygon Primitives → Sphere and open the option box.

To get started, first make sure History is turned on () in the status bar along the top of the UI), or there will be no creation node; then, click Create to make the poly sphere. Open the Attribute Editor and switch to its creation node, called polySpherel. In the creation node polySphere1 you'll find the Subdivisions Axis and Subdivisions Height sliders (in the option box, these are called Axis and Height Divisions), which you can use to change the surface detail retroactively.

The Polygon Tool

You use the Polygon tool to create a single polygon face by laying down its vertices (switch to the Polygons menu set and then choose Mesh Tools → EDIT → Create Polygon Tool, in the EDIT section of the menu). When you select this tool, you can draw a polygon face in any shape by clicking to place each point or vertex. Aside from creating a polygon primitive by choosing Create → Polygon Primitives, this is the simplest way to create a polygon shape. Figure 4.4 shows some simple and complex single faces you can create with the Polygon tool.

After you've laid down all your vertices, press Enter to create the poly face and exit the tool. For complex shapes, you may want to create more than just the single face so that you can manipulate the shape. For example, you may want to fold it.

Try This The poly shown in Figure 4.5 was created with the Polygon tool and has only one face. Therefore, adjusting or deforming the surface is impossible. To fold this object, you need more faces and the edges between them. Make your own intricate poly shape with the Polygon tool by clicking vertices down in the different views to get vertices in all three axes.

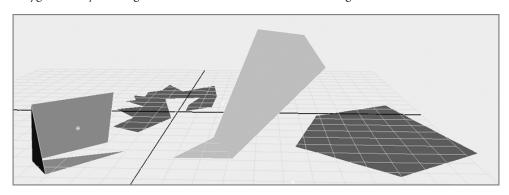
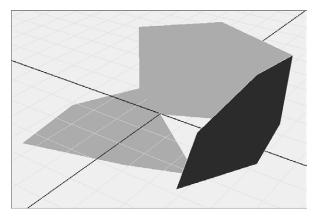


Figure 4.4 Polygon faces created with the Polygon tool

With the surface selected, choose Mesh - SHAPE - Triangulate. The surface has more faces and edges and is easier to edit, but it's still simple to create because you start with a single face. If you need a uniquely shaped poly, start with this tool and then triangulate your surface into several faces, as shown in Figure 4.6.



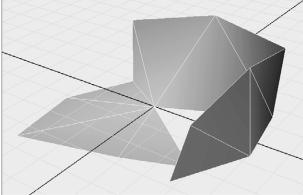


Figure 4.5 A single-faced polygon with a complex shape

Figure 4.6 Complex shapes are better with more faces.

Faces that have too many edges (a.k.a. Ngons) may cause you trouble later in the workflow, particularly in further manipulation of the mesh containing any Ngons and also when renderina.

Poly Editing Tools

Here's a brief preview of what to expect in the world of poly editing. You should experiment with each tool on a primitive sphere as it's introduced, so saddle up to your Maya window and try each tool as you read along.

Later in this chapter, you'll deploy these new skills. In Chapter 6, "Practical Experience," you'll create a cute toy airplane to exercise your modeling skills. For most of the work in this chapter, you'll use the Polygons menu. Open the Edit Mesh menu, tear it off, and place it somewhere on your screen so you can get a good look at the tools and functions, which are separated into sections according to function type. For example, tools that work on vertices are found under the Edit Mesh menu's VERTEX section. It's important to note these sections, as some tools have the same name and may be repeated more than once in the menu, but function differently if applied to vertices, faces, or edges.

Modeling Toolkit

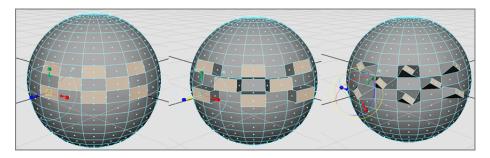
Modeling Toolkit integrates component-level selection and editing tools (such as selecting vertices, edges, and faces, and extruding them, for example) for a more streamlined

modeling workflow. Modeling Toolkit can make tedious modeling chores much easier, especially for advanced modeling techniques. I will be covering some of the Modeling Toolkit workflow and how it's integrated into Maya 2015 alongside Maya traditional workflows to give you a comparison and allow you to decide which workflow suits you. You'll take a look at Modeling Toolkit and its interface later in the chapter.

The Poly Extrusion Tools

The most commonly used poly editing tools have to do with extrusion. You can use Extrude to pull out a face, edge, or vertex of a polygon surface to create additions to that surface. You access the tool in the Edit Mesh menu under the respective section (FACE, EDGE, or VERTEX) - Extrude. Maya distinguishes between edge, face, or vertex extrusion based on which of those components you've selected and under which menu section you select the Extrude command. Follow these steps:

1. Select a face or multiple faces of a polygon and choose Edit Mesh → FACE → Extrude (under the FACE section of the menu). The regular manipulator changes to a special manipulator, as shown in the left image in Figure 4.7.



- 2. Grab the Z-axis move handle (the blue arrow) and drag it away from the sphere, as shown in the center of Figure 4.7.
- 3. Use the scale handles (the boxes) to scale the faces of the extrusion. The cyan circle rotates the face. The image at the right in Figure 4.7 shows the faces extruded, rotated, and scaled.
- 4. Choose the Extrude command again without deselecting the faces, and you extrude even more, keeping the original extrusion shape and building on top of that.
- 5. Select the edges of the poly surface instead of the faces and choose Edit Mesh → EDGE → Extrude (from the EDGE section this time) to extrude flat surfaces from the edges selected. The special manipulator works the same way as Extrude does for poly faces.

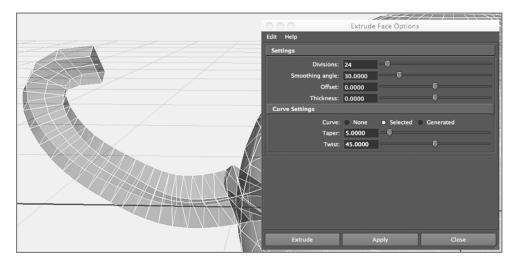
Figure 4.7

Extruding several faces at once on a sphere. The left image shows the selected faces, the middle image shows those faces extruded, and the right image shows those faces extruded with a rotation and smaller scale.

The faces you select pull out from the sphere, and new faces are created on the sides of the extrusion(s). The Extrude tool is an exceptionally powerful tool in that it allows you to easily create additions to any poly surface in any direction. Later in this chapter, you'll use it to make a simple cartoon hand.

You can also use the direction and shape of a curve to extrude faces. Create a curve in the shape you want your extrusion to take, select the curve, Shift+select the faces, and choose Extrude □. Taper decreases or increases the size of the face as it extrudes. Twist rotates the face as it extrudes, and Divisions increases the smoothness of the resulting extrusion. Choose Selected for the Curve setting. When you have your settings for those attributes, click the Extrude button (see Figure 4.8).

Figure 4.8 Extruding a face along a path curve



Although it seems to be strange behavior, the Twist and Taper values are taken into account in the extrusion. You can edit these values when you uncheck Selected, or you can reselect this option after you enter values for Twist and Taper. If your faces aren't extruding to the shape of the curve, increase the number of divisions.

Modeling Toolkit and Extrusions

Modeling Toolkit makes selecting and editing polygonal components more streamlined, accelerating some workflows by incorporating tools into one place for ease of access as well as by reducing how often you have to exit one tool or mode and enter another one. Since a lot of what Modeling Toolkit does centers around component selections, let's start there first.

Modeling Toolkit Interface

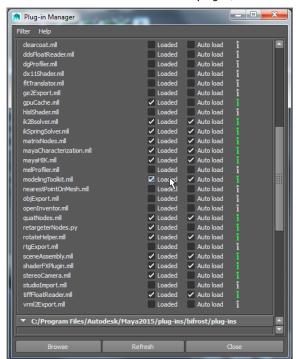
By default, the Modeling Toolkit plug-in should be enabled, which places the Modeling Toolkit menu on the main menu bar. If you don't see Modeling Toolkit, simply choose Window → Setting/Preferences → Plug-In Manager. About halfway down the list, you should see ModelingToolkit.dll (or ModelingToolkit.bundle on a Mac). Check Loaded and Auto Load, as shown in Figure 4.9.

Modeling Toolkit also places an icon on your status bar, next to the XYZ input fields, shown next to the cursor and already turned on in Figure 4.10. When the Modeling Toolkit icon is turned on, Modeling Toolkit is automatically invoked whenever you enter component selection mode. Click the icon to turn Modeling Toolkit on if it isn't already.

In addition, Modeling Toolkit places a tab in the Channel Box, called Modeling Toolkit, to make displaying its tool set easier, as shown in Figure 4.11. You will notice toward the top of the Modeling Toolkit panel four icons for selecting, moving, rotating, and scaling. These operate in the same way as transformation tools; however, they enable the Modeling Toolkit functionality. You'll see this in action throughout the book and introduced next.

Figure 4.9

Loading the **Modeling Toolkit** plug-in, if needed



Modeling Toolkit Extrusion

Now that you have a little background on how Modeling Toolkit integrates with Maya 2015, let's use it in comparison to the Maya Extrude tool you just used on a sphere.

- 1. Make sure the Modeling Toolkit icon (127) in the status bar is active to see the Modeling Toolkit tab alongside the Attribute Editor and Channel Box/Layer Editor tabs. Then make sure the Modeling Toolkit button in the upper-right corner of the Modeling Toolkit is activated () so that it is in blue.
- 2. Create a polygon sphere and press 5 for Shaded mode.
- 3. Right-click the sphere in your scene and select Face from the marking menu for face selection mode. This is the easiest way to select components in Maya, which also works the same while using Modeling Toolkit.



Figure 4.10 The Modeling Toolkit icon button

- 4. Your faces are now highlighting differently than they do in Maya, as you saw when modeling the decorative box in Chapter 3, "The Autodesk" Maya 2015 Interface." Now Modeling Toolkit selection turns the face dark red with a bold red highlight, instead of amber shading. Click a face to select it, or hold down Shift to select multiple faces. As soon as you click to select, the selected faces turn into the familiar Maya amber coloring while in Shaded mode. Select two faces side-by-side on the sphere.
- 5. In the Modeling Toolkit panel, click Extrude under the Mesh Editing Tools heading, as shown in Figure 4.12. The options appear below the button.
- 6. In the view panel, you'll see a yellow readout telling you the axis you will be extruding in, and your cursor will change to a double-headed horizontal arrow. Click and drag left or right to set the amount of extrusion (Figure 4.13).
- 7. In the Modeling Toolkit panel, click the Divisions button and set the number to 3. This will give you multiple sections along your extrusion. You can also click and drag in the view panel to set the Divisions number interactively.

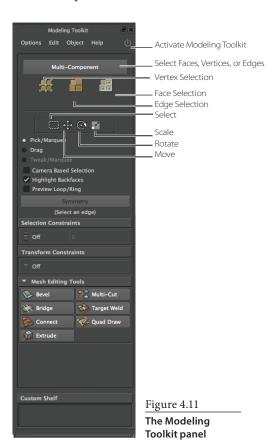




Figure 4.12 Click Extrude in the Modeling Toolkit panel.

- 8. Click the Offset button and click and drag in the viewport to make the extruded faces bigger or smaller. You may also enter a value in the Modeling Toolkit panel for Offset. Figure 4.14 shows an extrusion of 0.29 with a Divisions of 3 and an Offset of 0.04.
- 9. Finally, click the Keep Faces Together check box on and off in the Modeling Toolkit panel to see how the extrusion changes. Figure 4.15 shows the same

extrusion as Figure 4.14, but with Keep Faces Together turned off. Whatever options you set will be used the next time you extrude in Modeling Toolkit. Simply turn off the Extrude button to exit the tool and commit the changes.

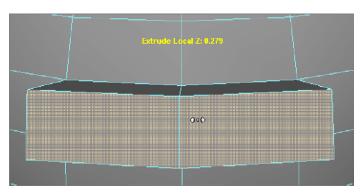


Figure 4.13 Click and drag to set the extrusion amount.

All of these extrusion options and settings are available in the Maya Extrude tool but are a little more streamlined in the Modeling Toolkit workflow. Experiment to see how you like to work. You will be using a combination of traditional Maya and Modeling Toolkit workflows throughout the chapter and other parts of the book.

The Wedge Tool

Similar to extruding faces, Wedge pulls out a poly face, but it does so in an arc instead of a straight line. For this tool, you need to select a face and an edge of the selected face for the pivot point of the corner. Here's how to do this. First deactivate Modeling Toolkit by clicking the power icon () to turn off the blue light.

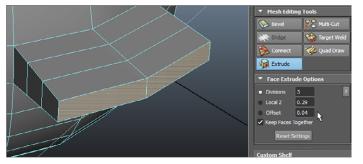


Figure 4.14 **Modeling Toolkit** extrusion in action

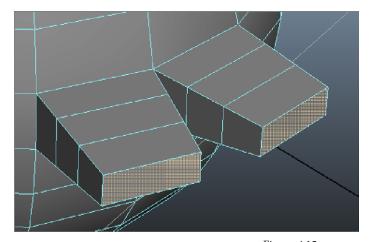


Figure 4.15 **Keep Faces Together** is turned off.

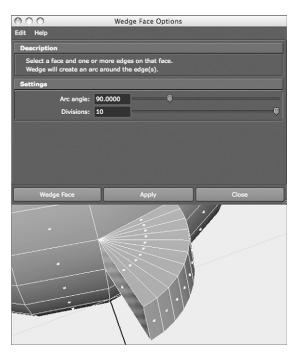
RMB+click a mesh and select Multi from the marking menu. Select a face, Shift+select one of the face's edges, and choose Edit Mesh → FACE → Wedge □ (under the FACE section of the menu).

Figure 4.16 **Executing a Wedge** operation on a face of a sphere

In the option box, notice the Description heading. Under the Settings heading, you can select the degree of turn in the Arc Angle (90 degrees is the default) as well as the number of faces used to create the wedge (by moving the Divisions slider), as shown in Figure 4.16.

As a reminder, you can RMB+click an object to display a marking menu. Drag the cursor to select Multi and release the mouse button to be able to select a face and then an edge for the Wedge operation. Then, click or Shift+click your selection.

The Wedge tool is useful for items such as elbows, knees, archways, and tunnel curves.



The Poke Tool

Poke is great for creating detailed sections of a mesh (poly surface) and bumps or indentations. To use the Poke tool to add detail to a face, select a face and then choose Edit Mesh → FACE → Poke.

A vertex is added to the middle of the face, and the Move manipulator appears on the screen for that new vertex, as shown in Figure 4.17. This lets you move the point to

your model.

Figure 4.17

Poke helps create areas of detail in

> where you need it on the face. You can add bumps and depressions to your surface as well as create regions of extra detail. By selectively adding detail, you can subdivide specific areas of a polygon for extra detailed work, leaving lower poly counts in less-detailed areas for an efficient model.

The Bevel Tool

Use the Bevel tool to round sharp corners and edges. The Bevel tool requires that you select an edge or multiple edges and then use them to create multiple new faces to round that edge or corner.

Select an edge or edges and choose Edit Mesh → EDGE → Bevel □ (under the EDGE section of the menu) to adjust your bevel. The Width slider sets the distance from the edge to the center of where the new face will be. This basically determines the size of the beveled corner. The Segments number defines how many segments are created for the bevel: The more segments, the smoother the beveled edge. Leaving Segments at 1 creates a sharp corner (see Figure 4.18).

The setting of the Roundness slider specifies the roundness of the corner. Setting the number too high will make the beveled edge stick out, as shown in Figure 4.19, although that can be a valid design choice. You can allow Maya to set the roundness automatically based on the size of the geometry being beveled. Select the Automatically Fit Bevel To Object check box to disable the Roundness slider. Move the Segments slider to set the number of new faces that are created on the bevel: The more segments, the smoother the bevel.

Use the Bevel tool to round polygonal edges. You can also use it to add extra surface detail because Bevel creates more faces on the surface.

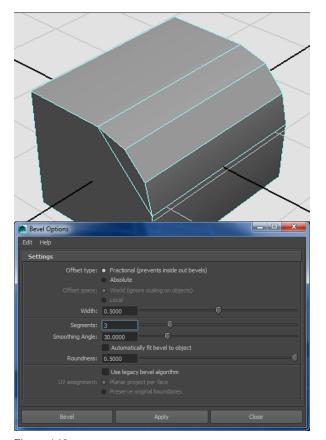


Figure 4.18 Increase Segments to create a rounder corner.

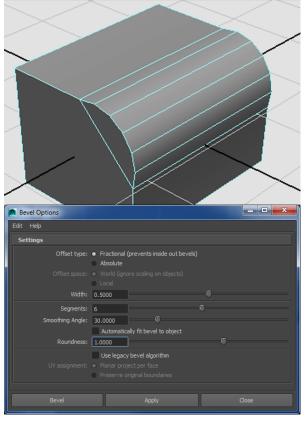


Figure 4.19 A poly bevel's roundness set too high

Modeling Toolkit Bevel

Just like the Maya Extrude and the Modeling Toolkit Extrude, there is a way to bevel inside Modeling Toolkit. Using the same example as earlier, a simple cube, you'll see how to bevel in the Modeling Toolkit here:

- 1. Make sure the Modeling Toolkit Activate icon () is turned on and enter edge selection mode.
- 2. Select an edge on the cube. Press W for the Move tool. You can also select Move from the Modeling Toolkit panel, but the traditional Maya hotkey of W works the same while in Modeling Toolkit. But because you have Modeling Toolkit enabled, your manipulator will have three circles instead of the squares from Maya's traditional Move manipulator. Just like with Maya's squares, you can click + drag any of the circles to see that you can move that edge in any two-dimensional plane easily. Moving your mouse over any of the circles gives you the plane axes (Figure 4.20). Undo any moves you may have done, and you'll move on to the Modeling Toolkit Bevel command next.

Figure 4.20 **Modeling Toolkit**

also has easy planar movement handles to components with its three circles in the manipulator.

- 3. With an edge selected, click the Bevel button in the Modeling Toolkit panel. You can select either Divisions or Offset and click and drag in the view panel to set their values. Offset is like Maya Bevel's Width attribute and makes the bevel larger or smaller, while Divisions affects the rounding of the bevel. Figure 4.21 shows a Modeling Toolkit bevel with an Offset of 0.5 and Segments of 3, which matches the Maya bevel done earlier with a Width of 0.5 and Segments of 3. The result on the cube is the same if you compare it with the earlier Maya bevel shown in Figure 4.18.
- 4. Click the Bevel button to exit the tool and commit your bevel operation to the cube.

As you can see, using Modeling Toolkit makes the Bevel tool slightly easier and faster to implement. As a matter of fact, Modeling Toolkit is only a workflow plug-in. It passes

all of its changes and work into standard Maya attributes and nodes, making sharing files created using Modeling Toolkit workflow no different from ones created without Modeling Toolkit enabled. So, the cube you bevel in Modeling Toolkit is precisely the same as the one beveled in Maya software's traditional workflow. The attributes and history on that object are the same.

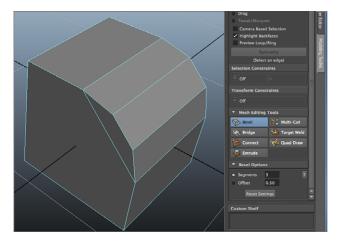


Figure 4.21

The Modeling Toolkit bevel is the same as the Maya bevel shown in Figure 4.18.

Having even a slightly rounded edge on a model—a box, for example—greatly enhances the look of that box when it's lighted and rendered, because the edges catch much more light, helping define the shape of the box. A perfectly sharp corner with no bevel doesn't catch any light, making the model look weaker.

Putting the Tools to Use: Making a Cartoon Hand

Starting with a simple polygonal cube, you'll create a basic cartoon hand using a mix of Maya and Modeling Toolkit workflows.

The book's web page (www.sybex.com/go/introducingmaya2015) as well as the author's website (www.koosh3d.com) will be hosting videos of some of the exercises in this book which may be updated and added to over time, so make sure to check in on these links every now and then.

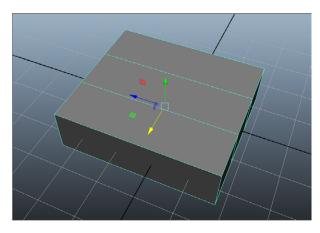
Download the entire Poly_Hand project from the web page (www.sybex.com/go/introducingmaya2015) where you can also find a video for this tutorial. Set your project to this folder and follow these steps:

1. With Interactive Creation turned off, create a polygonal cube. Open the Attribute Editor and, in the polyCubel tab, set Subdivisions Width to 3, Subdivisions Height

to 1, and Subdivisions Depth to 1. If you don't have that tab in the Attribute Editor, click Undo to remove the cube, turn on History, and re-create the cube.

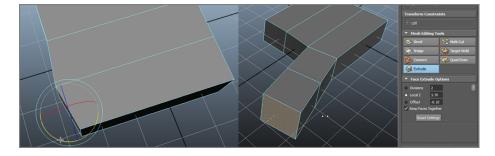
2. Scale the cube to X = 4, Y = 1.3, and Z = 4.5 so that it looks as shown in Figure 4.22.

Figure 4.22 The poly cube in position to make the hand



- 3. Turn on the Modeling Toolkit icon () in the status bar. Click the Modeling Toolkit tab in the Channel Box to see the Modeling Toolkit tools if you need them. RMB+click the cube and choose Face from the marking menu.
- 4. Select the front face that is in the corner closest to you. You'll extrude the face to make the first part of the index finger. Before you extrude, though, rotate the face a bit in the Y-axis, away from the rest of the hand, to angle the extrusion toward where the thumb would be, as shown in Figure 4.23 (left).

Figure 4.23 Rotate the face (left) and then extrude the index finger.



5. In the Modeling Toolkit panel, click the Extrude button. Set Divisions to 2. Click the Local Z radial button and enter 3.7 for the Extrude Local Z attribute in the Channel Box. Set the Offset to -0.10 and exit the Extrude tool. Figure 4.23 (right) shows the full index finger with the slight rotation away from the hand from the previous step.

Save your work, and compare it to the scene file poly_hand_v1.mb in the Poly_Hand project on the web page.

6. Repeat steps 4 and 5 for the remaining two fingers. Remember, if you rotate the initial face of each finger slightly away from the previous finger, the extrusions will have small gaps between them, as shown in Figure 4.24. Otherwise, the fingers will extrude right up against each other, like a glove with the fingers glued together. Also remember to exit the Extrude tool in the Modeling Toolkit panel by clicking it off after each extrusion is complete to move on to the next finger.

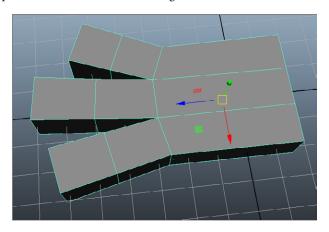


Figure 4.24 Three fingers

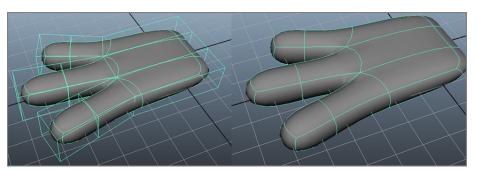
Use Table 4.1 as a guide for the extrusion lengths (Local Z value) for each finger.

When you're finished with the three fingers, select the hand; in the Perspective panel, press 2 to give you a smooth preview of the hand. With a polygonal object, pressing

FINGER	EXTRUDE LOCAL Z VALUE
Middle	4.2
Pinkie	3.0

Table 4.1 **Extrusion length** guide

the 1, 2, and 3 keys previews the smoothness your model will likely have when it's smoothed (a polygonal modeling operation about to be discussed). Pressing 2 also shows the original shape of the hand as a wireframe cage (see Figure 4.25, left).



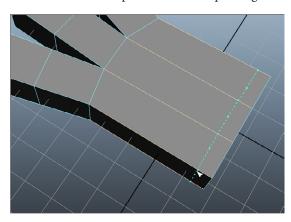
A smoothed preview of the hand, with the original shape shown as a cage (left) and a full smooth preview without the cage (right)

Figure 4.25

With the hand still selected, press 3. The original wireframe cage disappears, as shown in Figure 4.25 (right). This doesn't alter your model in any way; if you render,

- your hand will still be blocky, just as you modeled it. Press 1 to exit the smooth preview and return to the original model view. The scene file poly_hand_v2.mb shows the hand with the three fingers created.
- 7. Let's work on the thumb. You need to insert some new divisions on the body of the hand to extrude a thumb. With the hand selected, choose Mesh Tools → EDIT → Insert Edge Loop Tool and click the side edge of the hand, as shown in Figure 4.26. A dotted line appears. Drag along the edge you clicked to place the insertion and release the mouse button to place the new loop of edges.

Figure 4.26 Insert an edge loop at the base of the hand.



- 8. Insert a second loop of edges at the middle of the hand, as shown in Figure 4.27. Press W to exit the Insert Edge Loop tool.
- 9. Rotate and then scale down the face, as shown in Figure 4.28, in the Z-axis to get ready to extrude the thumb.

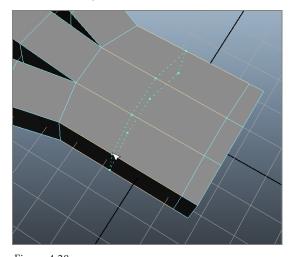


Figure 4.28 Scale and rotate the thumb's face to get it ready to extrude.

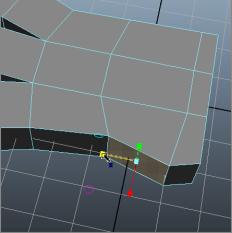


Figure 4.27 Insert a second loop of edges up toward the middle of the hand.

10. In Modeling Toolkit, click Extrude and set Local Z to 3.0. Keep Offset at -0.10 and Divisions to 2, as with the other three fingers (see Figure 4.29).

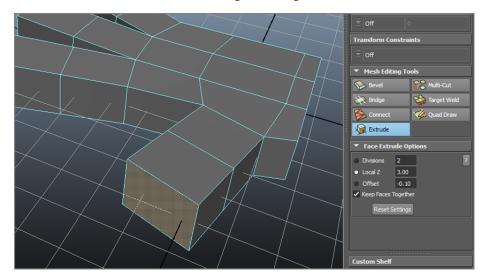


Figure 4.29 Extrude the thumb.

11. Select the three faces at the base of the hand (where it would meet an imaginary wrist) and scale them down in the X-axis and up in the Y-axis to create a flare, as shown in Figure 4.30.

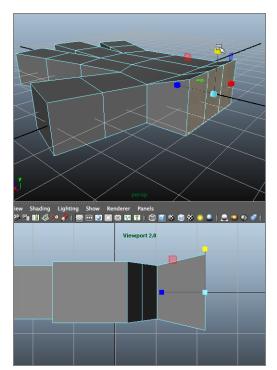


Figure 4.30 Create a flare at the base of the hand.

- 12. If you press 3 to preview the hand smoothed, it will lose a good amount of its detail and become too soft. Let's keep some of the angles of the hand by beveling the edges all around the hand. Select the outer edges all around the hand, as shown in Figure 4.31, in X-Ray view (which you invoke in a view panel by selecting in its menu bar Shading - X-Ray.). Make sure to get all the edges around the outside of the hand, fingers, and thumb.
- 13. In Modeling Toolkit, click Bevel and set Offset to **0.4** and Segments to **1**. Figure 4.32 shows the beveled hand without Smooth Preview (left) and with Smooth Preview (right).
- 14. While in Smooth Preview, select the ring of faces around the base of the thumb and scale them down a bit to accentuate the flare of the thumb, as shown in Figure 4.33. Make any additional adjustments to your liking by manipulating faces and vertices.

Figure 4.31 Select all the edges outlining the entire hand.

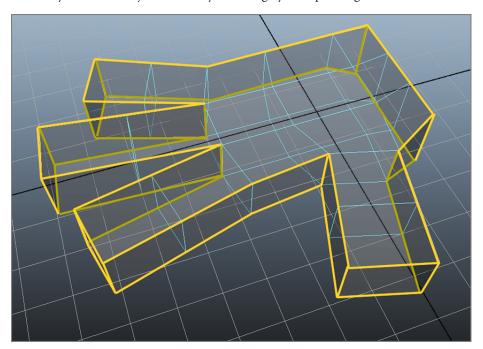
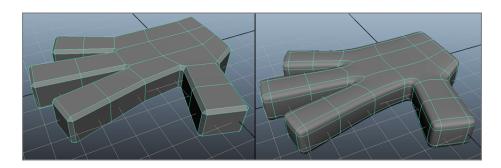


Figure 4.32 Beveling all around the hand (left) and shown with Smooth Preview (right)



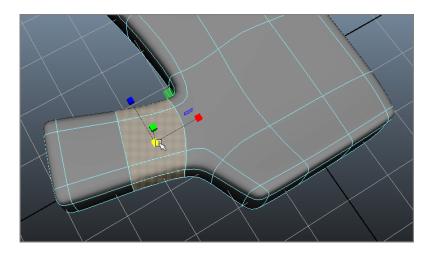


Figure 4.33 Scale down these faces to flare the thumb a bit more.

- 15. Exit Smooth Preview (press 1). To add more detail to the hand, you'll raise the knuckles. You need to create new vertices for the knuckles where each finger meets the hand. Enter edge selection mode with Modeling Toolkit still active. In the Modeling Toolkit panel, click the Multi-Cut icon (Click and drag along the edge of the hand, right below where the index finger attaches to the hand, until the yellow readout reads about 65%. Release the mouse button to lay down the first point of the Multi-Cut operation.
- 16. Now click the opposite edge across at about 33%. You'll notice a green dashed line stretch across denoting where a new edge will be placed, as shown in Figure 4.34 (left).

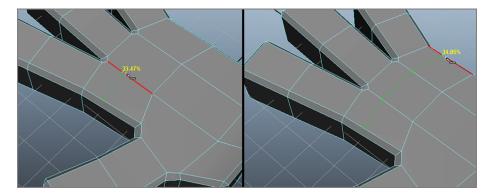
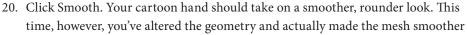


Figure 4.34 Use the Multi-Cut tool to lay down edges for the knuckles.

- 17. Click across the remaining knuckle faces to lay down a cut line across the top of the hand, as shown in Figure 4.34 (right).
- 18. Exit the Multi-Cut tool in the Modeling Toolkit panel to commit the changes, and it will add three new edges (and hence three new faces) along the back of the hand for the

- knuckles. Select each of those new faces and choose Edit Mesh → FACE → Poke (under the FACE section) to subdivide them into five triangles, with a vertex in the center. A special manipulator appears when you invoke the Poke command. Use the Z translate handle to pull those middle vertices up to make knuckles (see Figure 4.35).
- 19. Now that you have a cartoon hand, you can smooth out the mesh to make it less boxy. In Object mode, select the hand, and press 3 to see a preview of what the hand will look like after it's smoothed. Press 1 to exit the smooth preview. Choose Mesh → SHAPE → Smooth □ (under the SHAPE section of the menu); in the option box, under Exponential Controls, select Maya Catmull-Clark for the Subdivision Type, set Division Levels to 2, and leave the other options at their defaults (see Figure 4.36, left).

Figure 4.35 Use the Poke tool to raise the knuckles.



and given it a higher density of polygons. Notice all the nodes listed under Inputs in the Channel Box in Figure 4.36 (right). This is because History has been on for the entire duration of this exercise. At any time, you can select one of those nodes and edit something—the extrusion of the pinkie, for example.

To verify that you've been working correctly, you can load the finished hand file (with its history intact), which is called poly_hand_v3.mb, available from the book's web page, www.sybex.com/go/introducingmaya2015. If you don't need any of the history anymore, then with the hand selected, choose Edit → Delete By Type → History to get rid of all those extra nodes.

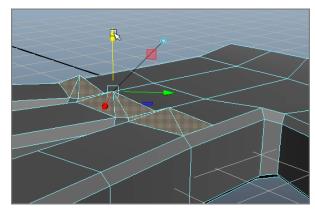
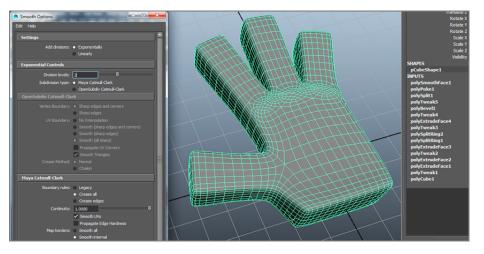


Figure 4.36 Set the options for the Smooth operation (left), and the smoothed hand is shown with all its history nodes (right).

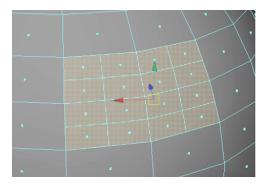


Creating Areas of Detail on a Poly Mesh

As you saw with the cartoon hand, it became necessary to add more faces to parts of the surface to create various details, such as with the knuckles. Maya provides several ways to add surface detail or increase a poly's subdivisions, as you've begun to see in the cartoon hand exercise. Let's take a deeper look at these and more tools for adding detail to a mesh.

The Add Divisions Tool

You can use the Add Divisions tool to increase the number of faces of a poly surface by evenly dividing either all faces or just those selected. Select the poly surface face or faces and choose Edit Mesh - FACE - Add Divisions (under the FACE section of the menu). Make sure not to select the first Add Divisions entry, which is under the EDGE section of the menu; otherwise, you will split the edges surrounding the selected face instead of splitting the face into more faces. In the option box, you can adjust the number of times the faces are divided by moving the Division Levels slider. With Add Divisions set to Exponentially under the Settings heading, the Mode drop-down menu gives you the choice to subdivide your faces into quads (four-sided faces, as on the left of Figure 4.37) or triangles (three-sided faces, as on the right in Figure 4.37).



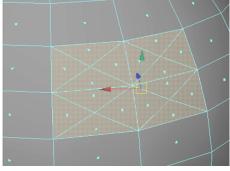


Figure 4.37

The Mode dropdown menu of the Add Divisions tool lets you subdivide faces into quads or triangles.

You can also select a poly edge to divide. Running this tool on edges divides the selected edges into separate edges along the same face, giving you more vertices along that edge. It doesn't divide the face; rather, you can use it to change the shape of the face by moving the new vertices or edge segments, as shown in Figure 4.38. Just make sure to select Add Divisions under the EDGE section of the Edit Mesh menu.

You use the Add Divisions tool to create regions of detail on a poly surface. This is a broader approach than using the Poke tool, which adds detail for more pinpoint areas.

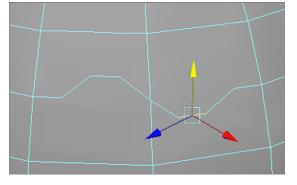
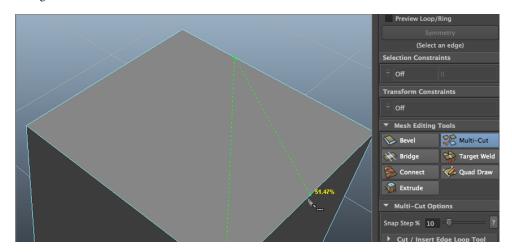


Figure 4.38 Dividing edges

Modeling Toolkit Multi-Cut Tool

As you saw when creating more faces and edges for the hand's knuckles in the previous exercise, Modeling Toolkit's Multi-Cut tool allows you to lay down edges along faces fairly easily. You access the Multi-Cut tool when Modeling Toolkit is enabled, under the Mesh Editing Tools heading. You can also make multiple cuts on the same face, as shown in Figure 4.39.

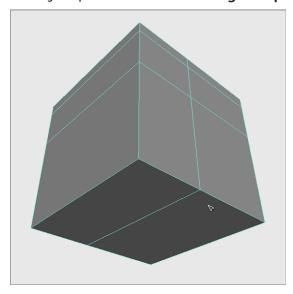
Figure 4.39 The Modeling **Toolkit** Multi-Cut tool



You can also access the Multi-Cut tool through Mesh Tools - EDIT - Multi-Cut Tool (under the EDIT section of the menu).

Figure 4.40 Using the Insert **Edge Loop tool**

The Insert Edge Loop Tool



This handy tool adds edges to a poly selection, much like the Multi-Cut tool, but it does so more quickly by working along the entire poly surface, along common vertices. The Insert Edge Loop tool automatically runs a new edge along the poly surface perpendicular to the subdivision line you click, without requiring you to click multiple times as with the Modeling Toolkit Multi-Cut tool. You used this tool in the decorative box in Chapter 3 and earlier in this chapter on the cartoon hand and will continue using it throughout this book. You'll find it indispensable in creating polygonal models because it creates subdivisions quickly.

For instance, subdividing a polygonal cube is quicker than using the Multi-Cut tool. With a poly cube selected, choose Mesh Tools - EDIT - Insert Edge Loop Tool. Click an edge, and the tool places an edge running perpendicular from that point to the next edge across the surface and across to the next edge, as shown in Figure 4.40. If you click and drag along an edge, you can interactively position the new split edges.

The Offset Edge Loop Tool

Much like the Insert Edge Loop tool, the Offset Edge Loop tool (Mesh Tools → EDIT → Offset Edge Loop Tool) inserts not one but two edge loop rings of edges across the surface of a poly. Edges are placed on both sides of a selected edge, equally spaced apart. For example, create a polygon sphere and select one of the vertical edges, as shown in Figure 4.41. Maya displays two dashed lines on both sides of the selected edge. Drag the mouse to place the offset edge loops; release the mouse button to create the two new edge loops.

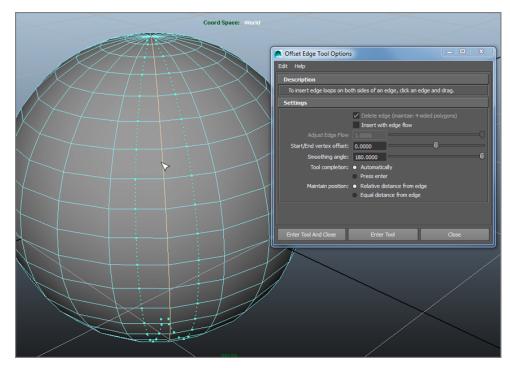


Figure 4.41 The Offset Edge Loop tool and its

options

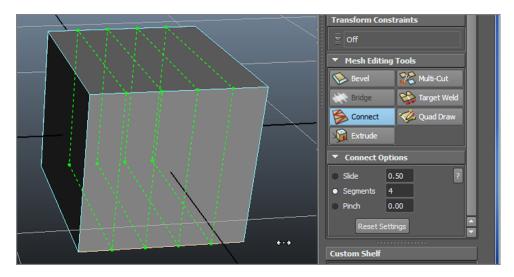
The Offset Edge Loop tool is perfect for adding detail symmetrically on a surface quickly.

Modeling Toolkit Connect Tool

Similar to the Insert Edge Loop tool is the Modeling Toolkit Connect function. While in Modeling Toolkit, simply select an edge and click the Connect button in the Modeling Toolkit panel. This will create edges going around the object perpendicular to the selected edge. The Slide attribute places the perpendicular cut along the selected edge, which is slightly less interactive than Insert Edge Loop. However, the Segments attribute

allows you to insert more than one ring of edges, while Pinch spaces those extra segments evenly (Figure 4.42). You can also select Mesh Tools - EDIT - Connect Tool in the main menu bar.

Figure 4.42 The Modeling **Toolkit Connect tool** creates edges much like Insert Edge Loop.



Modeling Toolkit Drag and Bridge Tool

One of Modeling Toolkit's nicest features is its drag selection mode. This allows you to essentially click and drag your cursor over the components you want selected with your cursor instead of having to click every component, almost like painting.

Try This

- 1. Create a cube in an empty scene and set Subdivisions Width to 1, Subdivisions Height to 4, and Subdivisions Depth to 5, as shown in Figure 4.43. Figure 4.44 is shown in X-Ray mode (which is enabled in the Perspective panel's menu bar by choosing Shading → X-Ray).
- 2. You are going to delete a square shape out of the front and back of the box. Exit X-Ray view mode if you are currently in X-Ray (view panel menu: Shading → X-Ray).
- 3. Make sure the Modeling Toolkit panel is open. Enter into face selection mode through the marking menu.
- 4. In the Modeling Toolkit panel, select the Drag option under the transformation icons, as shown in Figure 4.44. Your selection cursor changes to a circle. Click one of the inside faces, and drag along a six-face square in the middle of the front of the cube, also in Figure 4.44.

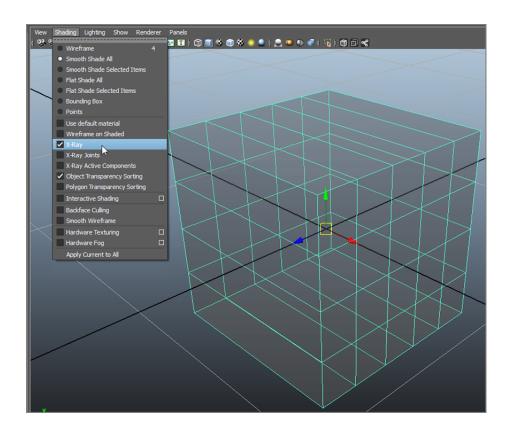


Figure 4.43 Create a subdivided box.

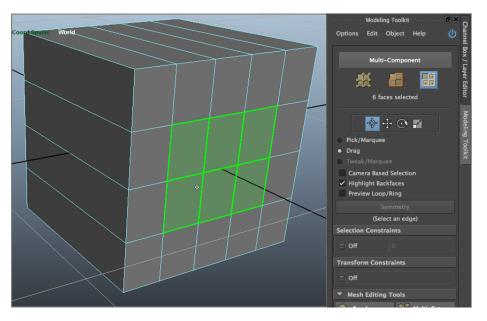
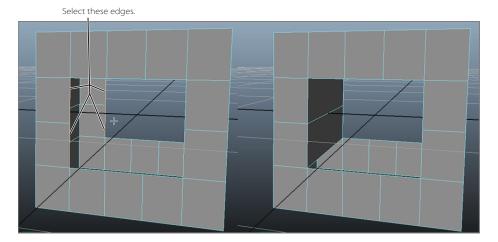


Figure 4.44 Drag+select a sixface square on the front of the box.

- 5. Orbit your view to see the back of the box, hold down Shift, and Drag+select the same six-face square on the back of the box.
- 6. Press Delete on your keyboard to delete the 12 selected faces, leaving you with a hollow box, as shown in Figure 4.45.
- 7. Now you're going to "fill in" the box to make a square-shaped tube. Switch to edge selection, and select the Pick/Marquee option in the Modeling Toolkit panel (it's right above the Drag option). Click to select the two front and back edges shown in Figure 4.46 (left).
- Figure 4.45 8. In the Modeling Toolkit panel, Delete the square shapes out of the box. click Bridge, and two new faces will appear connecting the selected edges (Figure 4.46, right). Exit the Bridge tool.

Figure 4.46 Select these edges (left) and bridge them (right).



- 9. Repeat steps 7 and 8 on the bottom three edges to connect the bottom, as shown in Figure 4.47.
- 10. Repeat for the remaining edges to fill the holes making a square pipe that is now solid on the inside.

Experiment with the Divisions attribute for the Modeling Toolkit Bridge to get a curvature in the bridged faces.

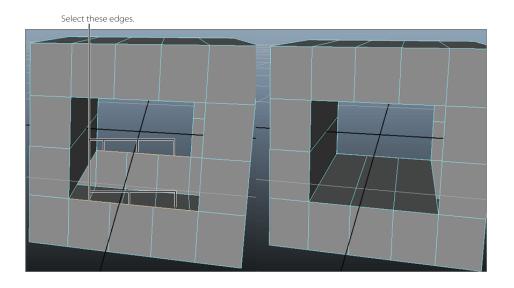


Figure 4.47 Bridge the bottom faces.

Modeling Toolkit Symmetry Selections

One of the charms of Modeling Toolkit is its ability to select in symmetry, meaning the components you select on one side of a surface are automatically selected on the other side, making modeling appreciably faster. While Maya has its own Symmetry feature in the transformation tools (Move, Rotate, Scale) covered in Chapter 3, it is limited to simple transforms. Tools such as Extrude or Bevel will not work in Maya's symmetry mode. Let's see how Modeling Toolkit Symmetry works.

Try This

- 1. Create a polygon sphere in a new scene. Press 5 for Shaded view, and make sure Modeling Toolkit is open.
- 2. Select an edge on the sphere that you want to be the centerline for the symmetry.
- 3. In the Modeling Toolkit panel, check the symmetry. Once you do, the sphere's object name will display next to the check box, as shown in Figure 4.48.
- 4. Now enter face selection. As you move your mouse before you select, it is mirrored, and when you do select a face or faces, that selection will be mirrored on the other side of the mesh (Figure 4.49).

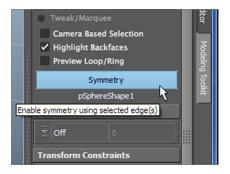
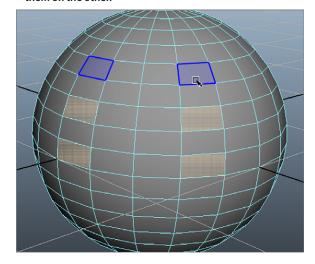


Figure 4.48 Turn on Symmetry mode.

Now if you engage any poly editing function, it will act on the symmetrically selected components.

Keep in mind that for Modeling Toolkit Symmetry to work, your mesh needs to be symmetrical itself. Uneven topology, where one side has a different number of faces than the other, will not work correctly.

Figure 4.49 Selecting faces on one side selects them on the other.



Combine, Merge Components, and the **Merge Vertex Tool**

The Combine function is important in cleaning up your model and creating a unified single mesh out of the many parts that form it. When modeling, you'll sometimes use several different polygon meshes and surfaces to generate your final shape. Using Combine, you can create a single polygonal object out of the pieces.

Frequently, when you're modeling a mesh, you'll need to fold over pieces and weld parts together, especially when you combine meshes into a single mesh. Doing so often leaves you with several vertices occupying the same space. Merging them simplifies the model and makes the mesh much nicer to work with, from rigging to rendering. The Merge Components tool fuses multiple

vertices at the same point into one vertex on the model. And the Merge Vertex tool from Modeling Toolkit is a more interactive way to fuse vertices together. We'll take a look at both in order to compare the workflows.

In the following simple example, you'll create two boxes that connect to each other along a common edge, and then you'll combine and merge them into one seamless polygonal mesh. To begin, follow these steps:

- 1. In a new scene, create two poly cubes and place them apart from each other, more or less as shown in Figure 4.50.
- 2. Select the bottom edge of the cube on the right that faces the other cube and choose Edit Mesh - EDGE - Extrude (in the EDGE section of the menu). Pull the edge out a little to create a new face, as shown in Figure 4.51. This will be a flange connecting the two cubes. It isn't important how far you pull the edge out; you'll connect the two cubes by moving the vertices manually.
- 3. Select the first corner vertex on the newly extruded face, and snap it into place on the corner vertex of the other cube, as shown in Figure 4.52 (left). Remember, you can click the Snap To Points icon () to snap the vertex onto the cube's corner.

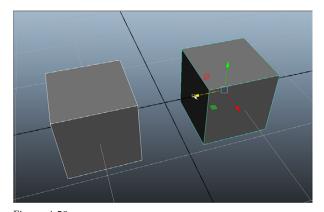


Figure 4.50 Place two polygonal cubes close to each other.

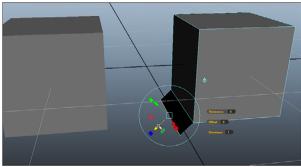


Figure 4.51 Extrude the bottom edge to create a flange.

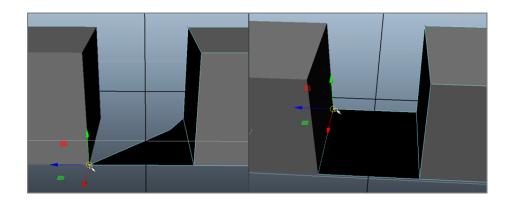


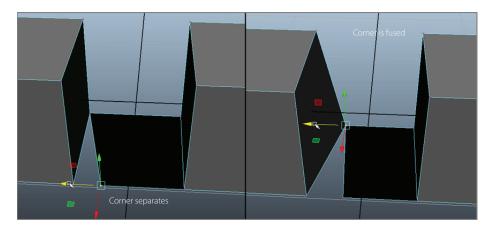
Figure 4.52 Snap the first corner vertex to the newly extruded face (left), and then the second (right).

- 4. Snap the other vertex to the opposite corner so that the cubes are connected with a flange along a common edge, as shown in Figure 4.52 (right).
 - Even though the cubes seem to be connected at a common edge, they're still two separate polygonal meshes. You can easily select and move just one of the stacked vertices and disconnect the connective face of the two cubes. You need to merge the stacked vertices of the cubes into a single vertex. However, the Merge Components function won't fuse vertices from two separate meshes together; you must first combine the cubes into a single poly mesh. The following steps continue this task.
- 5. Select the two cubes (one has the extra flange on the bottom, of course) and choose Mesh → COMBINE → Combine. Doing so makes a single poly mesh out of the two cubes. You can now use the Merge Components function.
- 6. Even though the cubes are now one mesh, you still have two vertices at each of the connecting corners of the cube on the left. Click on the vertex in the front corner of the newly combined boxes, and pull the vertex back as shown in Figure 4.53 (left) to

Figure 4.53

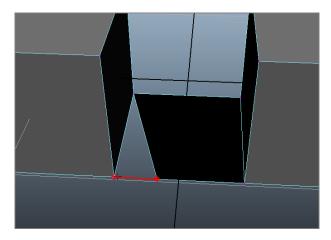
There are still two different vertices at the corner, and the boxes aren't really connected (left). The back corner is now connected properly (right).

disconnect the flange at one end. (Click the vertex to select just one. Don't use a marquee selection, which will select both vertices at once.)



- 7. Let's start at the back corner. To merge the vertices at that corner, select both the vertices that are on top of each other at the far corner by using a marquee selection and then choose Edit Mesh - Merge Components to fuse the two vertices sitting on top of each other together. As you can see in Figure 4.53 (right), clicking on the back corner vertex and moving it reshapes both cubes since the corner is now fused together.
- 8. Now for the front corner where we peeled back the vertex to create a gap in Step 6, this time using the Merge Vertex Tool instead. In this case, we do not need to have the two vertices sitting on top of each other or even close together as we need with Merge Components. Choose Mesh Tools - EDIT - Merge Vertex Tool. Click on the corner vertex on the flange on the cube on the right and then drag your cursor to highlight the corner vertex on the box on the left as shown in Figure 4.54. And now the near corner of this mesh is fused into a single vertex.

Figure 4.54 Merge Vertex tool in action.



To separate a combined mesh back into its component meshes, choose Mesh → SEPARATE → Separate. But you can't use Separate if the mesh you've combined has merged vertices.

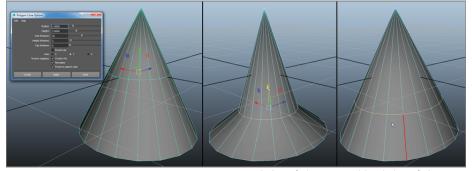
You'll notice fewer errors and issues with clean models when you animate, light, and render them. Combining meshes makes them easier to deal with, and Merge Components and the Merge Vertex tool cut down on unwanted vertices.

If the Merge Components function isn't working on vertices in your model, make sure the model is a single mesh, and if not, use the Combine function.

The Slide Edge Tool

If you need to move an edge on a model, selecting the edge or edges and using the Move tool will change the shape of the mesh. Let's see how this works:

1. Choose Create - Polygon Primitives - Cone □ and set the Height Divisions to 2 as shown in Figure 4.55 (left).



Moving the loop of edges

Sliding the loop of edges

Create a cone (left). The middle image shows what happens to the cone when you move the middle loop of edges as opposed to using the Slide Edge tool as shown on the right.

Figure 4.55

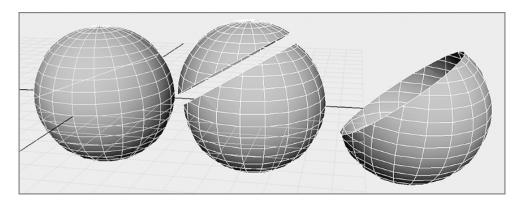
- 2. RMB+click and select Edges, and then double click on the middle row of horizontal edges to select the entire loop. Move the loop of selected edges down, and the cone turns into a pointy (Figure 4.55 [middle]).
- 3. Press Z for undo to return the object back to its original cone shape. Now, with the same loop of edges selected, choose Mesh Tools - EDIT - Slide Edge Tool and MMB+drag on one of the vertical edges (the vertical edge you MMB+click+ drag on will turn red) of the cone to slide the selected loop of edges up and down the cone without changing its shape, as shown in Figure 4.55 (right).

The Slide Edge tool is perfect for moving edges on a complex mesh surface without altering the shape of that surface.

The Cut Faces Tool

The Cut Faces tool lets you cut across a poly surface to create a series of edges for subdivisions, pull off a section of the poly, or delete a section (see Figure 4.56). Select the poly object, and choose Mesh Tools - EDIT - Cut Faces Tool. Click the option box if you want to extract or delete the section.

Figure 4.56 The Cut Faces tool can be used to create the edges, pull apart the poly object, or cut off a whole section.



You can use the Cut Faces tool to create extra surface detail, to slice portions off the surface, or to create a straight edge on the model by trimming off the excess.

The Duplicate Face Tool

Select one or more faces and choose Edit Mesh → FACE → Duplicate to create a copy of the selected faces. You can use the manipulator that appears to move, scale, or rotate your copied faces.

The Extract Tool

The Extract tool is similar to the Extrude tool, but it doesn't create any extra faces. Select the faces and choose Mesh - SEPARATE - Extract to pull the faces off the surface (see

Figure 4.57 Pull off the faces.

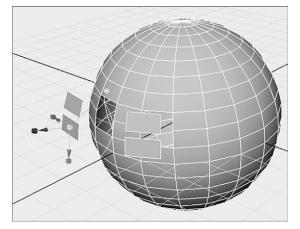


Figure 4.57). If the Separate Extracted Faces option is enabled, the extracted face will be a separate poly object; otherwise, it will remain part of the original.

This tool is useful for creating a new mesh from part of the original mesh you are extracting from. You can also use the Extract tool to create a hole in an object and still keep the original faces. When you use this tool with the Multi-Cut tool to make custom edges, you can create cutouts of almost any shape. You'll see this functionality of creating custom shaped holes with the Split Mesh with Projected Curve function explored below as well.

The Smooth Tool

The Smooth tool (choose Mesh - SHAPE - Smooth) evenly subdivides the poly surface or selected faces, creating several more faces to smooth and round out the original poly object, as you saw earlier in this chapter with the cartoon hand model exercise.

Creating a Hole in a Mesh Surface Using Split Mesh with **Projected Curve**

Sometimes you need to cut a hole in a mesh. You can simply select faces on that mesh and delete them to create a hole. However, if you need a more intricate, custom-shaped hole, you'll need to first use Split Mesh with Projected Curve to make a custom shape.

- 1. Create a polygon sphere in a new scene.
- 2. Now you'll create a curve directly on the sphere to outline the hole's shape. With the sphere selected, press the Make the selected object live icon in the status bar next to the snapping icons (). The text box next to this icon will change from stating No will enable you to perform some functions directly on that mesh.
- 3. Choose Create EP Curve Tool to draw a line. Click on the sphere and a little X appears. Click again and a line begins to appear. Click a number of points to create an interesting shape, and because the sphere is Live, these curve points will be placed on top of the sphere, as shown in Figure 4.58 (top). You can simply snap the final EP you draw on top of the first EP you drew with point snaps (\sum_1) to close the curve shape. In the Outliner the line is called curve1. More on creating curves can be found in Chapter 5.
- 4. We need to project this curved shape on to the sphere, because it is still a separate object despite you creating it on top of the sphere. First, disable Make Live by clicking its icon (again.

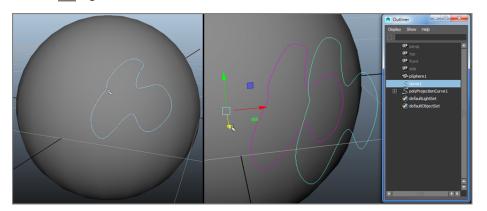


Figure 4.58

Draw an EP curve directly on the sphere (left). The projected curve (seen on the sphere in pink) adjusts if you move the original curve (seen away from the sphere in green).

- 5. Next, select the sphere and then the curved line and choose Edit Mesh → CURVE → Project Curve on Mesh. Now if you select curve1 in the Outliner (Figure 4.58, right) and move it around in your scene, the projected curve on the sphere (seen in your scene as pink) will adjust, staying on the sphere as shown in Figure 4.58 (right).
- 6. Now we can use this projected curve to outline a new set of edges on our sphere! In the Outliner, select the sphere and the projected curve (polyProjectionCurve1) and choose Edit Mesh → CURVE → Split Mesh with Projected Curve.
- 7. Maya creates a new sphere that is subdivided with the new edges you drew, as you can see in the Outliner in Figure 4.59 (left). You can delete the original sphere and curve you drew (pSphere1 and curve1), as well as the projected curve (polyProjectionCurve1), leaving you with pSphere2.

Figure 4.59 Splitting the sphere with its projected

curve (left), and then deleted those faces to make a custom hole (right).

> 8. Now simply delete the faces inside that shape, and you have your custom hole as shown in Figure 4.59 (right)!

Keep in mind creating custom holes like this in your mesh will likely create NGons, which are polygons with more than 4 sides. NGons are explained below in the catapult modeling exercise.

Projecting a line shape onto a mesh surface will allow you to not only cut holes as in this exercise, but it's also very handy for easily creating custom lengths of edges for modeling use.

The Sculpt Geometry Tool

You can use a Maya feature called Artisan to sculpt polygonal surfaces. Artisan is a painting system that allows you to paint attributes or influences directly onto an object. When you use Artisan through the Sculpt Geometry tool, you paint on a polygon surface to move the vertices in and out, essentially to mold the surface, as you will see with the

candle modeling exercise in Chapter 5, "Modeling with NURBS Surfaces and Deformers." In that chapter you will use this tool to add detail to a polygon model. Once you have played with the Sculpt Geometry tool in Chapter 5, try loading the poly_hand_v3.mb model and sculpting some detail into the cartoon hand.

To access the tool in polygon modeling, select your poly object and choose Mesh Tools → EDIT → Sculpt Geometry Tool □.

If you create a poly with a large number of subdivisions, you'll have a smoother result when using the Sculpt Geometry tool (see Figure 4.60).

Modeling a Catapult

You're going to create a catapult in this exercise using nothing but polygons. You'll use some sketches as a reference for the model. Since this is a more involved object than a hand, it's much better to start with good plans. This, of course, involves some research, web surfing, image gathering, or sketching to get a feel for what it truly is you're trying to make.

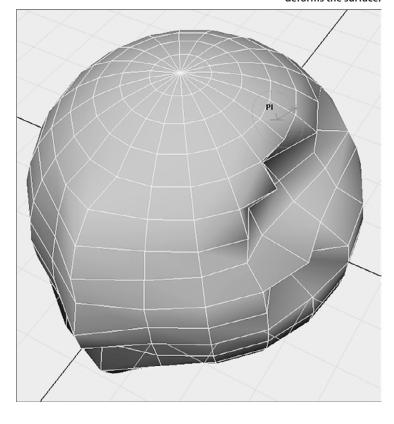
To begin, create a new project for all the files called Catapult, or copy the Catapult project from the companion website (www. sybex.com/go/introducingmaya2015) to your hard drive. If you do not create a new project, set your current project to the copied Catapult project on your drive. Choose File → Set Project and select the Catapult project downloaded from the companion website. Remember that you can enable Incremental Save to make backups at any point in the exercise.

Now, let's use a design already sketched out for reference. To begin, study the

design sketches included in the sourceimages folder of the project. These sketches set up the intent rather easily.

In Chapter 8, "Introduction to Animation," you'll animate the catapult. When building any model, it's important to keep animation in mind, especially for grouping related objects in the scene hierarchy so that they will move as you intend. Creating a good scene hierarchy will be crucial to a smooth animation workflow, so throughout this exercise you'll use the Outliner to keep the catapult's component pieces organized as you create them.

Figure 4.60 The Sculpt Geometry tool deforms the surface.



The Production Process

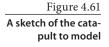
The trick with a complex object model is to approach it part by part. Deconstruct the major elements of the original into distinct shapes that you can approach one by one. The catapult can be broken down to five distinct objects, each with its own subobjects:

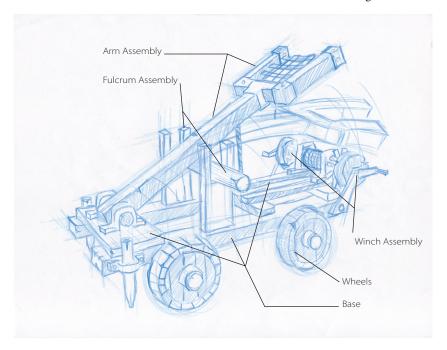
- Base
- Wheels
- Fulcrum assembly
- Winch assembly
- Arm assembly

You will model each part separately based on the sketch in Figure 4.61 and the detailed schematic in Figure 4.62.

The Base

The base consists of simple polygonal cubes representing timber and arranged to connect to each other. Keep in mind that in this exercise Interactive Creation for primitives is turned off (select Create - Polygon Primitives and make sure Interactive Creation is unchecked). Also, in the Perspective view, choose Shading - Wireframe On Shaded to turn on the wireframe lines while in Shaded mode to match the figures in this exercise.





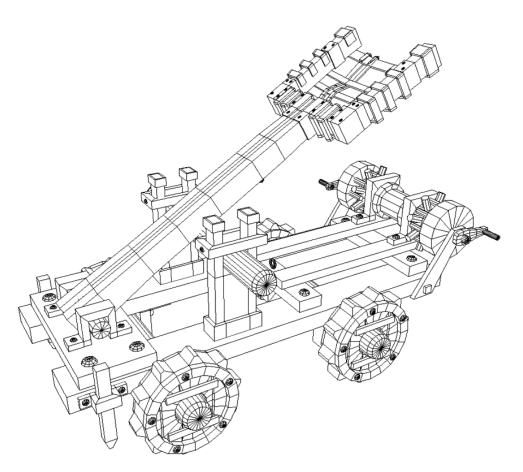


Figure 4.62 A schematic dia-

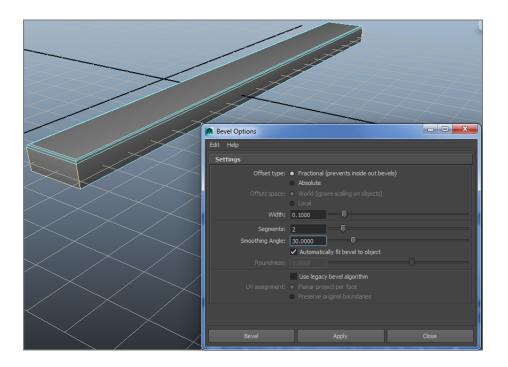
gram of the finished model to illustrate the goal

Creating the Base Objects

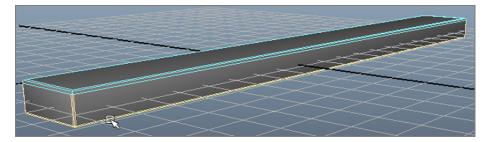
To begin the catapult base, follow these steps:

- 1. Choose Create → Polygon Primitives → Cube to lay down your first cube. This will be for the two long, broad boards running alongside.
- 2. Scale the cube to 2.0 in *X*, 0.8 in *Y*, and 19.5 in *Z*. Move it off the center of the grid about 2 units to the right.
- 3. Now you'll add some detail to the simple cube by beveling the sides, using either Modeling Toolkit or the traditional Bevel. Select the four edges running on top of the board. In the main menu bar, select Edit Mesh → EDGE → Bevel □. Set Width to 0.1, set Segments to 2, and click Apply. Figure 4.63 shows the resulting board.

Figure 4.63 Create a bevel for the baseboard object.



4. Select the remaining edges on the board, bevel them to a Width of **0.5**, and set Segments to 2. See Figure 4.64. If you use Modeling Toolkit, enter an Offset value of **0.5** and a Segments value of **2**.



Beveling the edges of your models can be an important detail. Light will pick up edges much better when they are beveled, even slightly. Perfect 90-degree corners can look too much like CG models and not real objects.

5. Select the board and choose Edit - Duplicate to place a copy of the board exactly where the original is in the scene. The new duplicated board is already selected for you, so just move the copy about 4 units to the left. You should now have something similar to Figure 4.65 (top).

Figure 4.64 Beveling the bottom edges

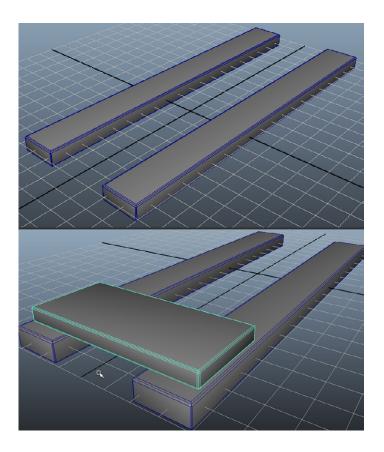
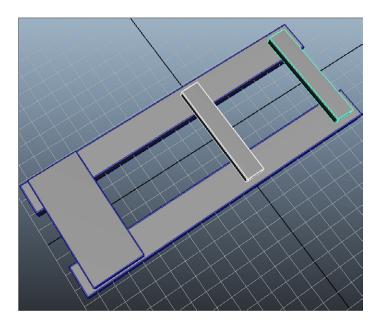


Figure 4.65 The long boards at the base (top) and

the platform board is in place (bottom).

- 6. Now for the cross braces and platform. Create a poly cube and scale it to 7.25 in *X*, 0.6 in Y, and 3.25 in Z. Place this platform on top of the two beams, at the end of the catapult's base.
- 7. With the first board that you beveled, you had a different bevel for the top edges than the bottom and sides. For this board, you'll have the same bevel width for all edges. Select the cube (not the edges as before) and choose Edit Mesh → EDGE → Bevel □. Set Width to **0.2**, set Segments to **2**, and click Bevel (or Apply). Figure 4.65 (bottom) shows the platform board in place and beveled.
- 8. Create a cube for the first of the top two cross braces, and scale it 6.5 in *X*, 0.6 in *Y*, and 1.2 in Z. Place it on top of the beams at the head of the base, and bevel this cube exactly as in the previous step.
- 9. Duplicate the cube, and move the copy about a third of the way down toward the end (Figure 4.66).

Figure 4.66 **Cross bracing** the base



Using Booleans

You're going to add some detail as you go along, namely, the large screws that hold the timber together. The screws will basically be slotted screw heads placed at the intersection of the pieces. In this section, you will use Booleans to help create the screw heads.

Booleans are impressive operations that allow you to, among other things, cut holes or shapes in a mesh fairly easily. Basically, a Boolean is a geometric operation that creates a shape from the addition of two shapes (Union), the subtraction of one shape from another (Difference), or the common intersection of two shapes (Intersection).

Be forewarned, however, that Boolean operations can be problematic. Sometimes you get a result that is wrong—or, even worse, the entire mesh disappears and you have to undo. Use Booleans sparingly and only on a mesh that is clean and prepared. You've cleaned and prepped your panel mesh, so there should be no problems. (Actually, there will be a problem, but that's half the fun of learning, so let's get on with it.)

First, you need to create the rounded screw head.

1. Create a polygonal sphere (Create → Polygon Primitives → Sphere) and move it from the origin off to the side in the *X*-axis of the base model. Scale the sphere down to 0.15 in *XYZ*.

2. With the sphere still selected, switch to the front view and press F to frame. RMB+click the sphere and select Face from the marking menu (Figure 4.67). Select the bottom half of the sphere's faces and press Delete on your keyboard to make a hemisphere (Figure 4.68).

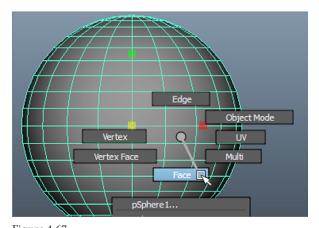


Figure 4.67

Use the marking menu to set the selection to Face.

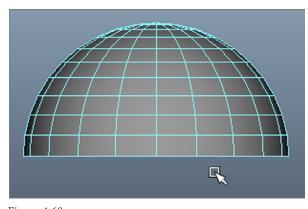


Figure 4.68 Delete the bottom half of the faces.

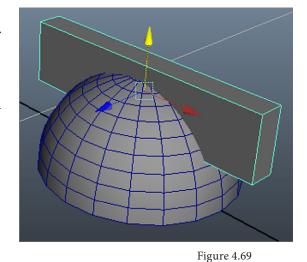
3. RMB+click the hemisphere and select Object Mode from the marking menu; this exits face selection mode. Create a poly cube and scale it to 0.4, 0.1, 0.04. Place it over the hemisphere as shown in Figure 4.69.

Now you have both objects that you need for a Boolean operation, and they are placed properly to create a slot in the top of the screw head.

4. Select the hemisphere and then the cube set into it. Select Mesh → COMBINE → Booleans → Difference (Figure 4.70). The cube disappears, and the screw head is left with a slot in the top, as shown in Figure 4.71.

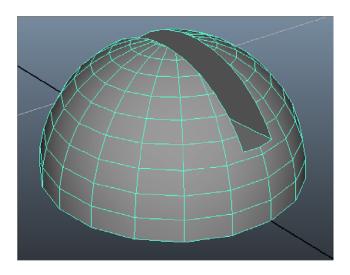


Figure 4.70 Selecting a Difference Boolean



Place the scaled cube over the screw head.

Figure 4.71 The screw head is slotted.

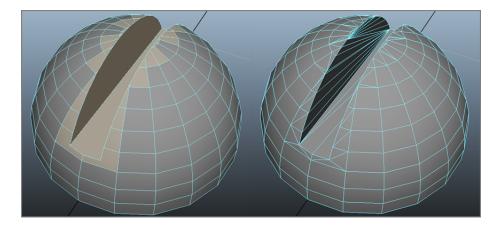


Ngons!

Now if you take a good close look at the screw head, especially where the slot is, you will notice faces that have more than four sides, which makes them Ngons. As I noted earlier in the chapter, faces that have more than four edges may be problematic with further modeling or rendering. This simple screw head most likely will not pose any problems in the application here, but let's go over how to prevent any problems early on. You will select the potential problem faces (those around the slot) and triangulate them.

1. Select all the faces around the slot, as shown in Figure 4.72 (left). Choose Mesh → SHAPE - Triangulate. This is the easiest and fastest way to subdivide these faces from being Ngons without having to use the Multi-Cut tool to manually fix them. And although it may not look as clean as before (Figure 4.72, right), the geometry is clean and will not be a potential problem like Ngons would be.

Figure 4.72 Select the faces around the slot (left) and triangulate them (right).



- 2. Select the screw head, and choose Edit → Delete By Type → History. This cleans out any history on the object now that you're satisfied with it.
- 3. Notice that the screw head's pivot point is at the origin. With the object selected, choose Modify → Center Pivot.
- 4. Name the object ScrewHead and position it at one of the intersections of the boards you've built so far.
- 5. Duplicate that first screw head and place the copies one by one at all the other intersections on the base, as shown in Figure 4.73. These are pretty big screws, huh? For this simple catapult, they'll do fine. The workflow to make more realistic screws is the same if you want to make this again with more realism and scale.

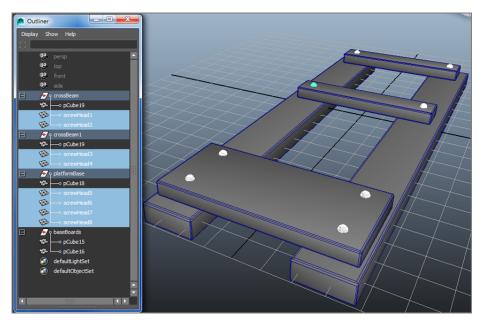


Figure 4.73 Place the screw heads on the base and organize your scene.

6. Now take the objects in the scene and group them into a logical order, as shown in the Outliner in Figure 4.73.

Save your file, and compare it to catapult_v1.mb in the Catapult project from the companion website to see what the completed base should look like.

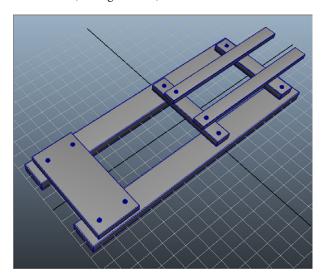
The time you spend keeping your scene objects organized now will pay off later when you animate the catapult in Chapter 8.

The Winch Baseboards

Next to model for the base are the bars that hold the winch assembly to the base. Refer to the sketch of the catapult (Figure 4.60, earlier) to refresh yourself on the layout of the catapult and its pieces. Follow these steps:

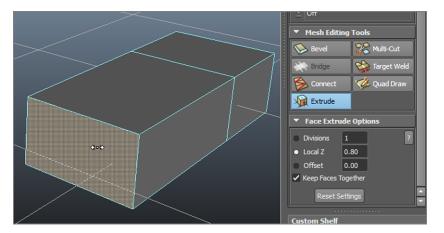
1. Create two long, narrow, beveled poly cubes for the baseboards of the winch, and position them across the top two side braces. Put a couple of screws on the middle crossbeam (see Figure 4.74).

Figure 4.74 Adding the winch baseboards



2. For the brackets that hold down the winch, create a small poly cube and move it off to the side of the base to get it out of the way. Scale the cube to 0.5, 0.3, 0.45. Enable Modeling Toolkit, select the side face, and click the Extrude button in the Modeling Toolkit panel. Type **0.8** for the Local Z attribute and click the Extrude button off to commit the extrusion, as shown in Figure 4.75.

Figure 4.75 **Use Modeling** Toolkit Extrude to extrude the face.



- 3. Select the top face of the original cube and use the Modeling Toolkit Extrude tool to extrude it to a Local Z of **1.54** to take it up to an L shape. Select the two inside vertices on the top of the L and move them up to create about a 45-degree angle at the tip, as shown in Figure 4.76.
- 4. Select that angled face and select Edit Mesh → FACE → Extrude (in the FACE section of the menu) for a Maya extrusion. Click the cyan-colored switch icon above the Extrude manipulator (shown next to the cursor in Figure 4.77, left). This will switch the extrusion axis (Figure 4.77, center) so you can pull the faces out straight and not angled up. Then grab the Z Move manipulator and pull the extrusion out about 0.75 units.

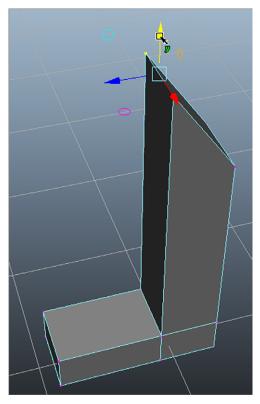


Figure 4.76 Extrude the top out to create an L shape: then move the vertices up to angle the top of the L.

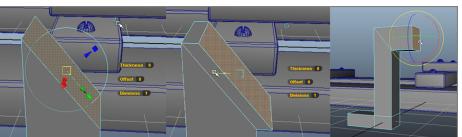


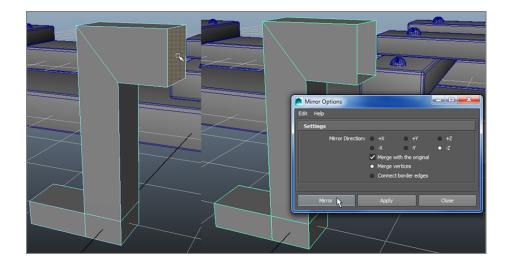
Figure 4.77

Click the switch icon (left) to switch the axis of extrusion (center). Rotate and scale the face to square it (right).

- 5. Press W to exit the Extrude tool and enter Rotate. Select the end face and rotate it to make it flat vertically and scale it down in Y-axis to prevent it from flaring upward (Figure 4.77, right).
- 6. This shape forms half of the braces you need. To create the other half, select the face shown in Figure 4.78 (left) and delete it (press Delete). Enter Object mode, select the mesh, and choose Mesh → MIRROR → Mirror Geometry □. Set Mirror Direction to -Z, and leave the options as shown in Figure 4.78 (right). Click Mirror, and you will have a full bracket.

Figure 4.78

Delete the face (left) and set the Mirror Geometry options (right).



- 7. Name the object **bracket**, and move it on top of one of the baseboards for the winch; then place a duplicated screw head on the flanges of the bracket. Group the bracket and screw heads together by selecting them and choosing Edit → Group; call the group **bracketGroup**.
- 8. Duplicate bracketGroup, and move the copy to the other baseboard, as shown in Figure 4.79. Organize your scene as shown in the Outliner in Figure 4.79.

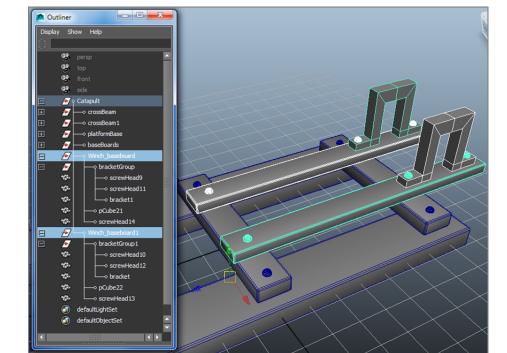


Figure 4.79
The winch's base completed

The Ground Spikes

The last items you need for the base are the spikes that secure the base into the ground at the foot of the catapult. Follow these steps:

- 1. Duplicate a bracket group and name it bracket-**GroupCOPY**. Remove the group from its current hierarchy (the baseboard group) by MMB+dragging it to another location in the Outliner (see Figure 4.80). Center its pivot (Modify → Center Pivot).
- 2. Move the bracket to the other side of the base. Rotate it on its side, scale it to about half its size in all three axes, and place it as shown in Figure 4.81 (left). Select the top vertices and move them closer to the base, as shown in Figure 4.81 (right).
- 3. Now for the spike itself. Create a poly cube and position and scale it to fit through the bracket. Scale the spike cube to about 3.5 in the Y-axis. Select the bottom face of the spike cube and choose Edit Mesh -

FACE - Extrude. Click the Thickness helper and type in a Thickness of **0.5** and an Offset of 0.15, as shown in Figure 4.82. Click in an empty area to commit the extrusion. You can also use Modeling Toolkit to create the extruded spike.

♠ Outliner

Display Show Help

60

go.

@

@

₽

ð

TEN TO

THE REAL PROPERTY.

1600 A

1600 A

15th

2

-o crossBeam

-o crossBeam1

- platformBase

-o baseBoards -o Stake

→ Winch_baseboard

o nCube22

defaultLightSet

defaultObjectSet

o screwHead13

o bracketGroup1

--o bracket

-o screwHead9

-o screwHead11

⊸ Stake1

##

#####

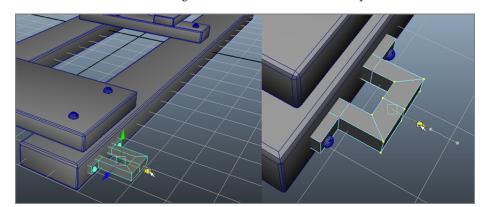


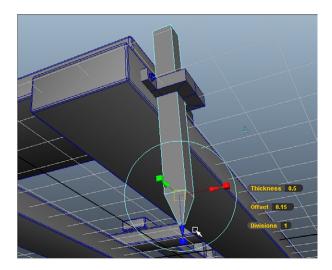
Figure 4.80

MMB+dragging the duplicated bracketGroup to another location in the **Outliner removes** the group from the Winch_baseboard1 group.



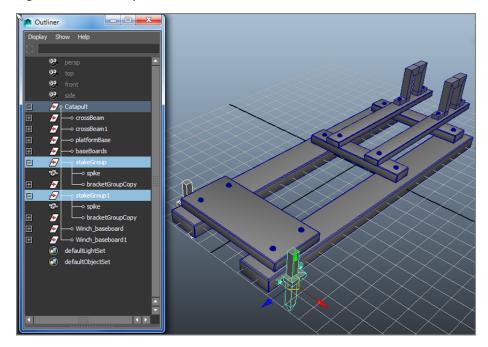
Position and scale the bracket assembly for the ground spikes (left). Move the vertices to reduce the depth (right).

Figure 4.82 Creating the spike



- 4. Bevel the spike if you'd like. Then select spike and bracketGroupCOPY and group them together, calling the new group stakeGroup; center its pivot.
- 5. Duplicate the stake group and move and rotate it 180 degrees in the *Y*-axis to fit to the other side of the base. Organize everything into a parent Catapult group (see Figure 4.83) and save your scene as a new version.

Figure 4.83 The completed base



The scene file catapult_v2.mb in the Catapult project from the companion website has the completed base for comparison.

The Wheels

What's a catapult if you can't move it around to vanquish your enemies? So now, you will create the wheels. Follow these steps:

1. First is the axle. Create a polygon cylinder (Create → Polygon Primitives → Cylinder) and then scale, rotate, and place it as shown in Figure 4.84 to be the rear axle.

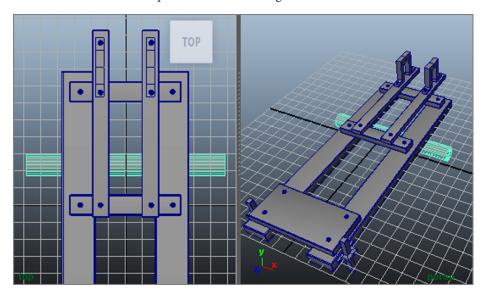


Figure 4.84 Place the rear axle.

- 2. Duplicate one of the stake assembly's bracket groups (bracketGroupCOPY) two times; then move and scale each of the two copies to hold the axle on either side.
 - Move down the top vertices of the bracket to make the bracket fit snugly around the axle as needed, as in Figure 4.85. Remember to move the duplicated brackets out of their existing hierarchy in the stakeGroups. Group both the axle brackets together and name the group Axle_Brackets. You are not grouping the brackets with the axle cylinder. Keep them separate. You'll organize the hierarchy better a little later.
- 3. To make the axle a little more interesting, let's add a taper at the ends. You will insert new edges around the ends by using Insert Edge Loop, which will be much faster than the Multi-Cut tool in this case. Select the rear axle cylinder and choose Mesh Tools → EDIT → Insert Edge Loop Tool. Your cursor turns into a triangle. Select one of the horizontal edges on the cylinder toward one end, as shown in

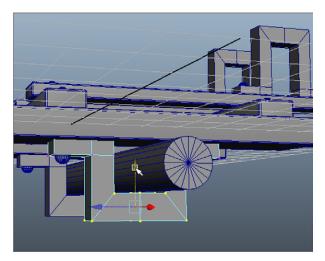


Figure 4.85

Place brackets to hold the rear axle and adjust the vertices to make it fit.

- Figure 4.86. A dashed line will appear running vertically around the cylinder. Drag the cursor to place the dashed line as shown in Figure 4.86 and release the mouse button to commit the new edges to that location. Repeat the procedure for the other side.
- 4. Select the end cap faces and scale them down on each side of the axle cylinder, as shown in Figure 4.87, to create tapered ends. Name the cylinder rearAxle. Now you're ready for the rear wheels.

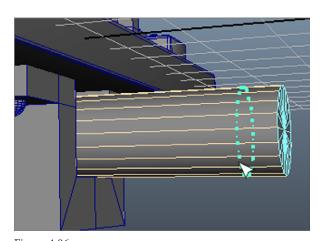


Figure 4.86 Insert an edge loop around the end of the cylinder.

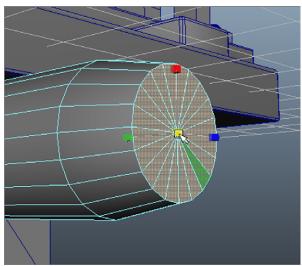
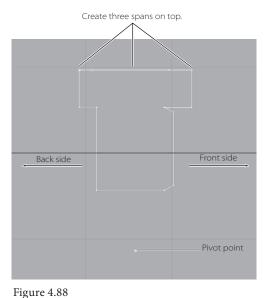


Figure 4.87 Taper the ends of the axle.

5. To model a wheel, first you'll use NURBS curves to lay out a profile to revolve. Go into the front view. Choose Create → CV Curve Tool □ and select 1 Linear for Curve Degree. Since the wheel's profile will have no smooth curves, you can create a linear CV curve like that in Figure 4.88. It's important for the design to create three spans for the top part of the curve. Place the pivot point (hold down the D key, or press Insert on a PC or Home on a Mac) about 3/4 of a unit below the curve, as shown. This curve will be the profile of the front of the wheel.



The profile curve for the wheel

6. Place the profile above the rear axle. To make sure the pivot point for the profile lines up with the center of the axle, turn on Snap To Points (a.k.a. Point Snap) () and press and hold down D to move the pivot. Snap the pivot to the center of the axle, as shown in Figure 4.89. Turn off Point Snap.

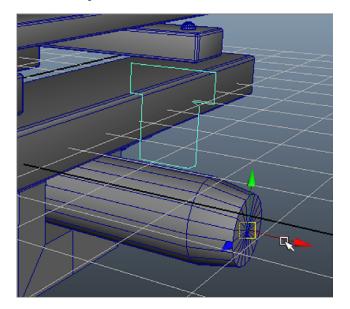


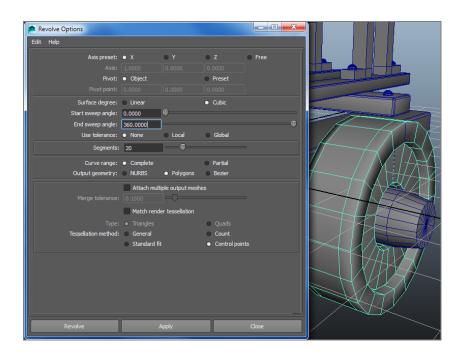
Figure 4.89 The profile curve is in place for the rear wheel.

- 7. Switch to the Surfaces menu set (Figure 4.90). Select the curve and revolve it by choosing Surfaces → Revolve □. In the option box, set Axis Preset to X to make it revolve correctly. Change Segments from the default 8 to **20** to give a smoother wheel. Set Output Geometry to Polygons, and set Tessellation Method to Control Points. This will create the edges of the faces along the CV points on the curve. Click Revolve, and there it is (Figure 4.91).
- 8. Select the wheel object and bevel it. With the wheel still selected, delete the history and the original NURBS curve since you won't need either again.
- 9. Add some detail to the wheel. Duplicate a screw head and remove the copy from whatever group it was in by MMB+dragging it out of the current group in the Outliner. Arrange a few of the screw heads around the front face of the wheel.



Figure 4.90 Switch to the Surfaces menu set.

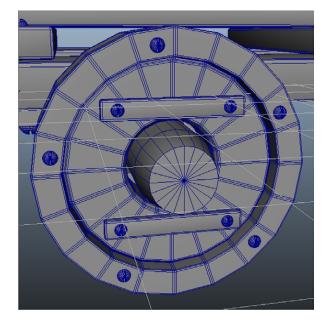
Figure 4.91
The wheel revolved



10. Add a couple of braces on the front of the wheel above and below the wheel's middle hole with two thin, stretched, and beveled poly cubes, with screws on either side (as shown in Figure 4.92). Again, make sure to remove the duplicated screw heads from whatever group you got them from.

Figure 4.92

Adding detail to the wheel



- 11. Select all the objects of the wheel, group them together by pressing Ctrl+G, and call the group **wheel**. Center the wheel group's pivot point by choosing Modify → Center Pivot.
- 12. Adding studs to the wheel makes for better traction when moving the catapult through mud and also for a cooler-looking catapult. To create all the studs at once, grab every other middle face along the outside of the wheel and extrude them with a Thickness of -0.3 and an Offset of 0.1, as shown in Figure 4.93. You can use Modeling Toolkit Extrude instead to make the same extrusions if you prefer.

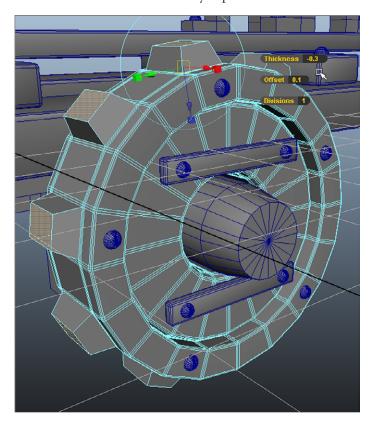


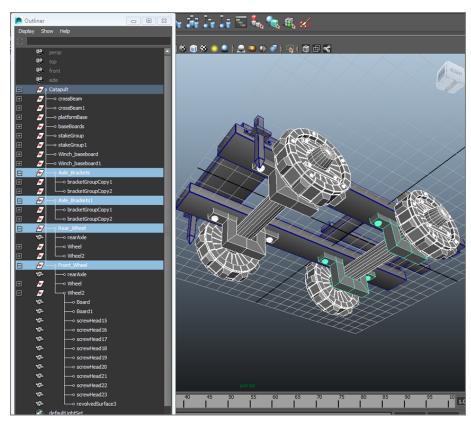
Figure 4.93 Extrude out studs for the wheel.

- 13. Copy the wheel group and rotate it 180 degrees in the Y-axis to create the other rear wheel for the other side. Position it on the other side of the rear axle.
- 14. Group the two wheels with the rear axle and call the new group node **Rear_Wheel**.
- 15. Select the Rear_Wheel node and the Axle_Brackets group node, and duplicate them by choosing Edit - Duplicate or by pressing the hotkey Ctrl+D. Move the objects to the foot of the catapult for the front wheels. Rename the wheel group node Front_Wheel.

16. Add the new axle bracket and wheel group nodes to the Catapult top node by MMB+dragging them onto the Catapult node in the Outliner; save your scene. Figure 4.94 shows the positions and Outliner hierarchy of the wheels.

Figure 4.94

The wheels and brackets are positioned, and the hierarchy is organized.



The file catapult_v3.mb in the Catapult project from the companion website reflects the finished wheels and base.

The Winch Assembly

To be able to pull the catapult arm down to cock it to fire a projectile, you'll need the winch assembly to wind a rope that connects to the arm to wind it down into firing position. Since animating a rope can be a rather involved and advanced technique, the catapult will not actually be built with a rope. To build the winch assembly, follow these steps:

1. The first part of the winch is the pulley around which the rope winds. In the front view panel, create a profile curve for extrusion as you did with the wheel that looks more or less like the profile curve in Figure 4.95. In this figure, the first CV of the profile curve is on the left end of the curve. Place the pivot point of the curve at that first CV. Revolve the curve around the X-axis with only 12 segments (as opposed to the wheel's 20). Center its pivot, and you have the pulley.

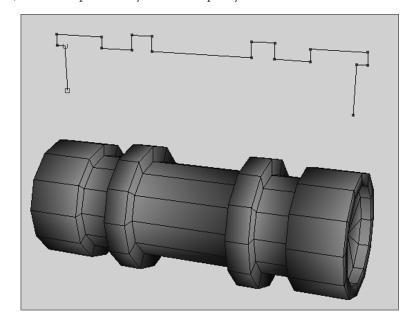


Figure 4.95

Create a profile curve and revolve it to create the object seen below the profile curve.

Figure 4.96 Place the pulley.

- 2. Position the pulley at the rear of the catapult, placing the brackets in the grooves (see Figure 4.96).
- 3. Now you'll need some sort of geared wheel and handle to crank the pulley. Create a poly cylinder, and rotate it so it's on its side like one of the wheels. Scale it to a squat disk with scale values of **1.4** in the *X*- and *Z*-axes and **0.4** in the *Y*-axis. Select the disk and bevel it.
- 4. Off on the side of your scene, create another poly cylinder, and rotate it to its side as well. Scale it to be a long, thin stick. You'll use this as the first of eight gear teeth for the wheel. Position it at the top of the wheel, as shown in Figure 4.97. Click the Snap To Points icon in the Status line () and snap the pivot point (press D to move the pivot) to the center of the wheel. Turn off Snap To Points.

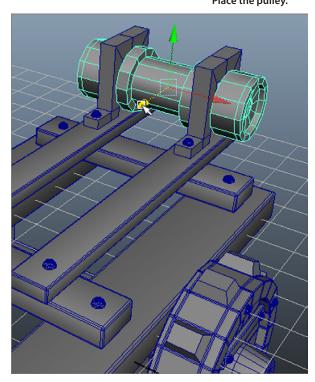


Figure 4.97

Making a gear

5. Instead of duplicating the gear tooth and positioning it seven more times, you'll use the array capabilities of the Duplicate Special tool. Select the tooth and choose Edit → Duplicate Special □. In the option box, set Rotate to **45** in the *X*-axis, and set Number Of Copies to 7. Since the pivot for the tooth is at the center of the wheel, as soon as you click the Duplicate Special button, Maya places seven copies around the wheel at 45-degree intervals (Figure 4.98).

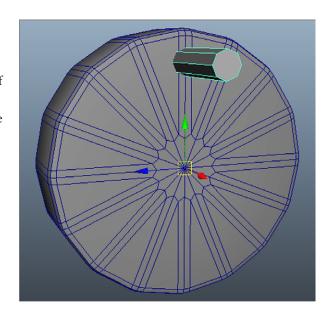
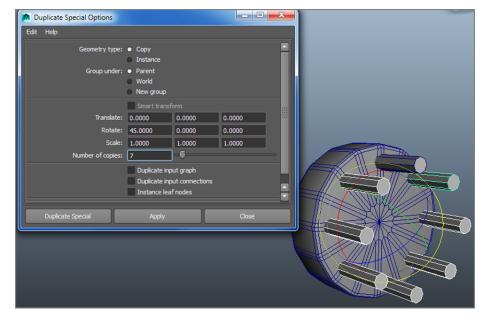


Figure 4.98

Eight gear teeth in place



6. Now for the handle. Create a poly cube with enough segments for you to adjust vertices and faces to match the handle shown in Figure 4.99. Create cylinders for the crank axle and handle and place them as shown. Group all the parts together and

snap the pivot point to the center of the gear wheel disk. Name the group handle. You can bevel the handle if you want.

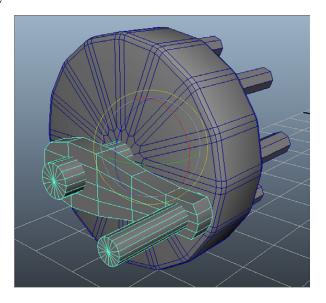


Figure 4.99

Use two cylinders and a poly cube to create the handle shapes.

7. Group the geometry together, call the object Turn_Wheel, and center its pivot. Place it at the end of the pulley. Place a copy (rotated 180 degrees) on the other side of the pulley. Figure 4.100 shows the placement.

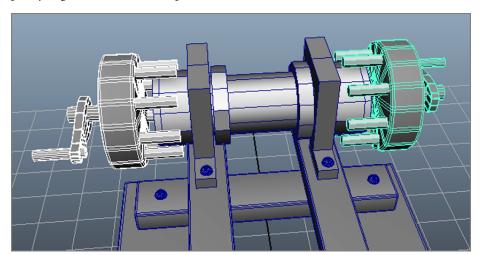


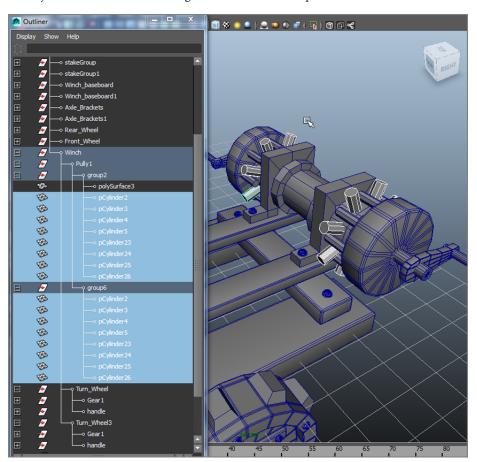
Figure 4.100

Place the turn wheels.

8. Now you'll need gear teeth on the pulley cylinder shape. Create a poly cylinder to be a long, thin tube like the gear teeth, and position it at the end of the pulley. Place it so that it is in between two of the turning wheel gear teeth. Place the pivot at the center of the pulley using Snap To Points.

- 9. Duplicate the new tooth seven times around the pulley at 45-degree intervals with Duplicate Special.
- 10. Make a copy of each of those eight teeth and move the copies to the other side of the pulley for the other gear. Group the pulley and turning wheels together, and name the object **Winch**, as shown in Figure 4.101. Center the pivot.

Figure 4.101 The winch gears and handles



11. Using a couple of poly cubes that you shape by moving vertices, make a winch arm on either side to brace the winch to the catapult. Bevel the shapes when you are happy with their shapes. Place the braces between the crank handle and the turning wheel on both sides, and bolt them to the catapult's base, as shown in Figure 4.102. Group them and add them to the hierarchy as shown. Save your scene file.

To verify your work up to this point, compare it to catapult_v4.mb in the Catapult project from the companion website.

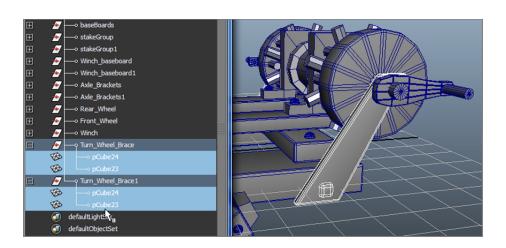


Figure 4.102

The assembled winch

The Arm

OK, now I'm kicking you out of the nest to fly on your own! Try creating the arm (see Figure 4.103), without step-by-step instruction, using all the techniques you've learned and the following hints and diagrams:

• Create the intricate-looking arm with face extrusions. That's all you'll need for the arm geometry. Follow Figure 4.104 for subdivision positions to make the extrusions work correctly.

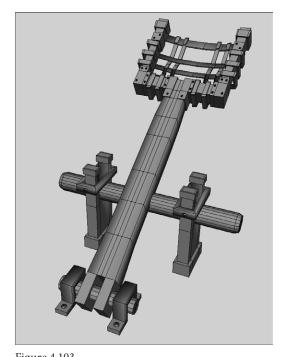


Figure 4.103 The catapult arm assembly

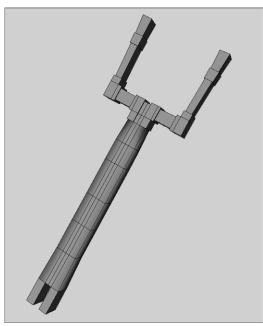
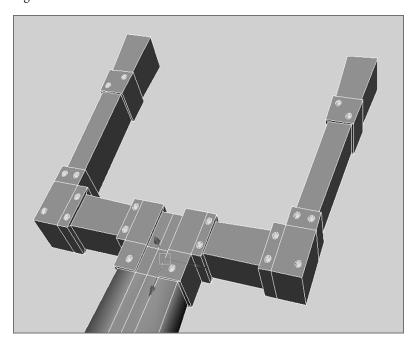


Figure 4.104 Follow the subdivisions on your model.

Figure 4.105 Place screw heads around the

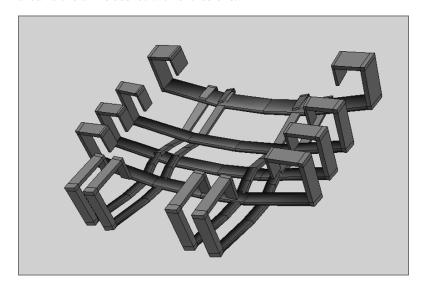
basket arms.

• Duplicate and place screw heads around the basket assembly, as shown in Figure 4.105.



• Create the straps for the basket with poly cubes. It's easier than it looks. You'll just need to create and extrude the cubes with enough subdivisions to allow you to bend them to weave them together, as shown in Figure 4.106. The ends of the straps wrap around the arm's basket with extrusions.

Figure 4.106 **Basket straps**



- Create the hinge for the arm with a couple of duplicated brackets and a cylinder.
- Create the arm's stand with multiple extrusions from a cube. Follow the subdivisions in Figure 4.107 for reference.

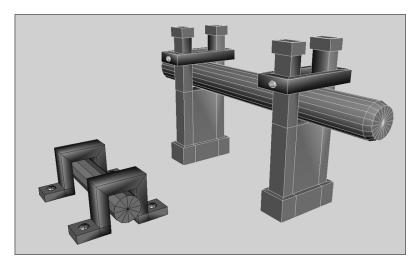


Figure 4.107 Follow the subdivisions on the arm stand.

- Bevel the parts you feel could use some nice edging, including the arm and stand pieces.
- Group the objects together and add their groups to the Catapult node.

When you've finished, save your scene file and compare it to catapult_v5.mb in the Catapult project from the companion website. Figure 4.108 shows the finished catapult.

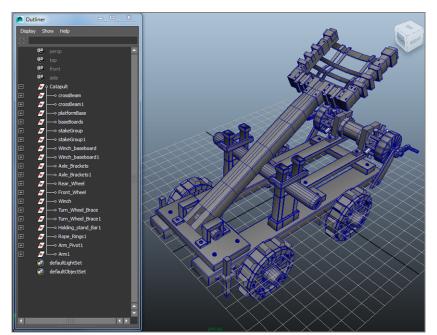


Figure 4.108 The completed catapult

Summary

In this chapter, you learned about the basic modeling workflows with Maya and Modeling Toolkit and how best to approach a model. This chapter dealt with polygon modeling and covered several polygon creation and editing tools, as well as several polygon subdivision tools. You put those tools to good use by building a cartoon hand and smoothing it out, as well as making a model of an old-fashioned catapult using traditional Maya workflows as well as new Modeling Toolkit workflows. The latter exercise stressed the importance of putting a model together step-by-step and understanding how elements join together to form a whole in a proper hierarchy. You'll have a chance to make another model of that kind in Chapter 6, when you create a toy airplane that is used to light and render later in the book.

Complex models become much easier to create when you recognize how to deconstruct them into their base components. You can divide even simple objects into more easily managed segments from which you can create a model.

The art of modeling with polygons is like anything else in Maya: Your technique and workflow will improve with practice and time. It's less important to know all the tricks of the trade than it is to know how to approach a model and fit it into a wireframe mesh.



Modeling with NURBS Surfaces and Deformers

 $NURBS\ is\ based\ on$ organic mathematics, which means you can create smooth curves and surfaces. NURBS models can be made of a single surface molded to fit, or they can be a collection of patches connected like a quilt. In any event, NURBS provides ample power for creating smooth surfaces for your models.

Now that you've learned the basics of creating and editing poly meshes, you'll get into creating some organic surfaces using NURBS techniques. This chapter also explains how to use deformers to adjust a model, as opposed to editing the geometry directly as you did with the previous modeling methods.

Learning Outcomes: In this chapter, you will be able to

- Use the surfacing techniques (Loft, Set Planar, and Revolve) to create surfaces
- Convert NURBS geometry into polygons
- Create polygon meshes directly from NURBS techniques
- Model a pair of glass candle holders and convert them to polygons
- Create CV curves and use Revolve to create a poly mesh
- Attach and detach curves
- Use the Sculpt Geometry Tool to sculpt a mesh surface
- Use the Soft Deformation Modifier
- Use nonlinear deformers to adjust existing geometry
- Create edits to existing models using lattices
- Animate an object to deform through a lattice

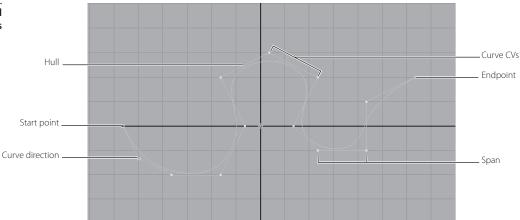
NURBS for Organic Curves

NURBS is an acronym for Non-Uniform Rational B-Spline. That's good to know for cocktail parties. NURBS modeling excels at creating curved shapes and lines, so it's most often used for organic forms such as animals and people, as well as highly detailed cars. These organic shapes are typically created with a quilt of NURBS surfaces, called patches. Patch modeling can be powerful for creating complex shapes such as characters, but it can also be quite difficult and will not be covered in this book. I will however, touch on the basics of NURBS surface modeling in this chapter.

In essence, Bézier curves are created with a starting and ending *control vertex* (CV) and usually two or more CVs in between that provide a smooth curvature. As each CV is laid down, the curve or spline tries to go from the previous CV to the next one in the smoothest possible manner.

As shown in Figure 5.1, CVs control the curvature. The hulls connect the CVs and are useful for selecting multiple rows of CVs at a time. The starting CV appears in the Autodesk® Maya® software as a closed box. The second CV, which defines the curve's direction, is an open box, so you can easily see the direction in which a curve has been created. The curve ends, of course, on the endpoint CV. The start and end CVs are the only CVs that are always actually on the curve itself.

Figure 5.1 A Bézier curve and its components



NURBS Modeling

NURBS surfaces are defined by curves called *isoparms*, which are created with CVs. The surface is created between these isoparms to form spans that follow the surface curvature defined by the isoparms, as in Figure 5.2. The more spans, the greater the detail and control over the surface.

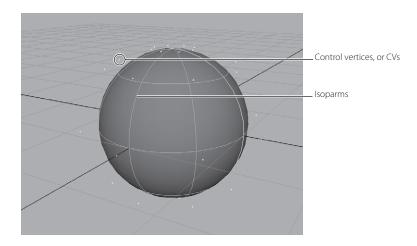


Figure 5.2 **NURBS** surfaces are created between isoparms. You can sculpt them by moving their CVs.

It's easier to get a smooth deformation on a NURBS surface with few CVs than on a polygon. Achieving the same smooth look on a polygon would take much more surface detail. As you can see in Figure 5.3, NURBS modeling yields a smoother deformation, whereas polygons can become jagged at the edges.

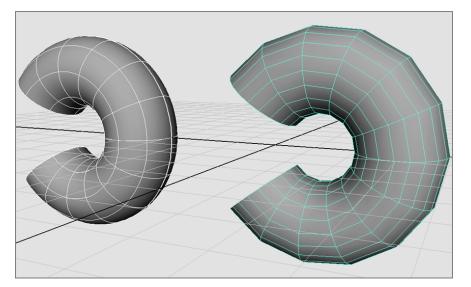


Figure 5.3

A NURBS cylinder (left) and a polygonal cylinder (right) bent into a C shape. The NURBS cylinder remains smooth, and the polygon cylinder shows its edges.

You can convert NURBS to polygons at any time, but reverting to NURBS can be tricky.

Try This Open a new scene (choose File → New Scene). In the new scene, you'll create a few curves on the ground plane grid in the perspective (persp) panel. Maximize the perspective view by moving your cursor to it and pressing the spacebar. Choose Create → CV Curve Tool. Your cursor turns into a cross. Lay down a series of points to define a curved line on the grid. Notice how the Bézier curve is created between the CVs as they're laid down.

Spans are isoparms that run horizontally in a NURBS surface; sections are isoparms that run vertically in the object.

NURBS surfaces are created by connecting (or spanning) curves. Typical NURBS modeling pipelines first involve creating curves that define the edges, outline, paths, and/ or boundaries of surfaces.

A surface's shape is defined by its isoparms. These surface curves, or curves that reside solely on a surface, show the outline of a shape much as the chicken wire does in a wire mesh sculpture. CVs on the isoparms define and govern the shape of these isoparms just as they would regular curves. Adjusting a NURBS surface involves manipulating the CVs of the object.

Levels of Detail

Figure 5.4 A loft created with four curves that are selected in order from left to right

NURBS is a type of surface in Maya that lets you adjust its detail level at any time to become more or less defined as needed. Pressing 1, 2, or 3 toggles between detail levels for any selected NURBS object from low quality at 1 to high quality at 3.

NURBS Surfacing Techniques

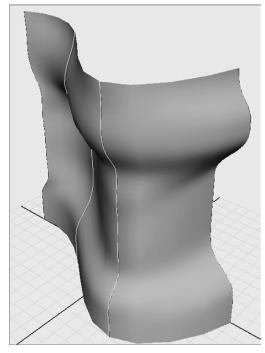
The easiest way to create a NURBS surface is to create a NURBS primitive, and then you can sculpt the primitive surface by moving its CVs. But you can also make surfaces in several ways without using a primitive. All these methods involve using NURBS curves to define a boundary, shape, or path of the surface and then using one of the methods described in the following sections to create the surfaces.

Lofting

The most common surfacing method is *lofting*, which takes at least two curves and creates a surface span between each selected curve in the order in which they're selected. Figure 5.4 shows the result of lofting a few curves together.

To create the loft in Figure 5.4, follow these steps:

- 1. Switch to the Surfaces menu set (press F4).
- 2. Draw the two curves.



- 3. Select the curves in the order in which you want the surface to be generated.

When you define more curves for the loft, Maya can create more complex shapes. The more CVs for each curve, the more isoparms you have, and the more detail in the surface. Lofting works best when curves are drawn as cross-sectional slices of the object to be modeled.

Lofting is used to make a variety of surfaces, which may be as simple as tabletops or as complex as human faces.

Revolved Surface

A revolved surface requires only one curve that is turned around a point in space to create a surface, like a woodworker shaping a table leg on a lathe. First you draw a profile curve to create a profile of the desired object, and then you revolve this curve (anywhere from 0 degrees to 360 degrees) around a single point in the scene to sweep a surface.

The profile revolves around the object's pivot point, which is typically placed at the origin but can be easily moved. Figure 5.5 (left) shows the profile curve for a wine glass. The curve is then revolved around the Y-axis a full 360 degrees to create the wine glass. Figure 5.5 (right) is the complete revolved surface with the profile revolved around the Y-axis.

To create a revolved surface, draw and select your profile curve and then choose Surfaces → Revolve.

A revolved surface is useful for creating objects such as bottles, furniture legs, and baseball bats anything that is symmetrical around an axis.

Extruded Surface

An extruded surface uses two curves: a profile curve and a path curve. First you draw the profile curve to create the profile shape of the desired surface. Then you sweep the curve from one end of the path curve to its other end, creating spans of a surface along its travel. The higher the CV count on each curve, the more detail the surface will have. An extruded surface can also take the profile curve and simply stretch it to a specified distance straight along one direction or axis, doing away with the profile curve. Figure 5.6 shows the profile and path curves, and Figure 5.7 shows the resulting surface after the profile is extruded along the path.

Figure 5.5 A profile curve is drawn in the outline of a wine glass in the Y-axis (left image) and the revolved surface

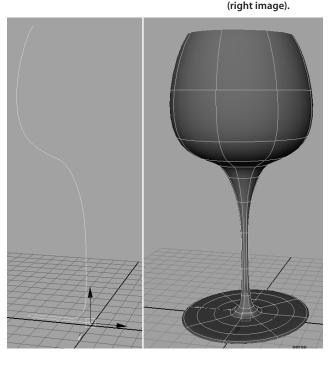


Figure 5.6

The profile curve is drawn in the shape of an I, and the path curve comes up and bends toward the camera.

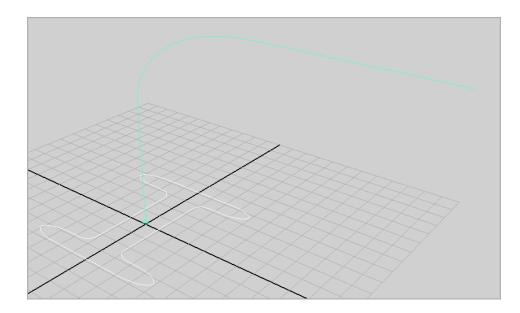
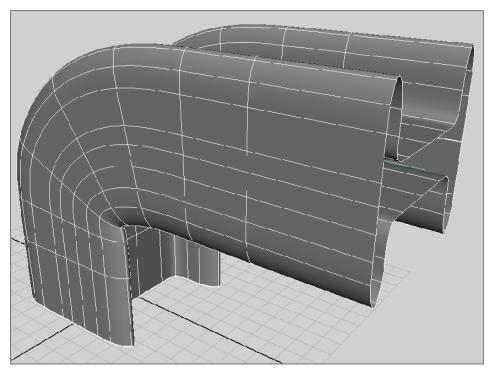


Figure 5.7

After extrusion, the surface becomes a bent I-beam.



To create an extruded surface, follow these steps:

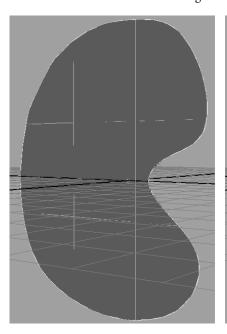
- 1. Draw both curves.
- 2. Select the profile curve.
- 3. Shift+click the path curve.
- 4. Choose Surfaces → Extrude.

An extruded surface is used to make items such as winding tunnels, coiled garden hoses, springs, and curtains.

Planar Surface

A planar surface uses one perfectly flat curve to make a two-dimensional cap in the shape of that curve. You do this by laying down a NURBS plane (a flat, square NURBS primitive) and carving out the shape of the curve like a cookie cutter. The resulting surface is a perfectly flat, cutout shape.

To create a planar surface, draw and select the curve and then choose Surfaces → Planar. You can also use multiple curves within each other to create a planar surface with holes in it. A simple planar surface is shown on the left side of Figure 5.8. When a second curve is added inside the original curve and both are selected, the planar surface is created with a hole. On the right side is the result when the outer curve is selected first and then the inner curve is selected before choosing Surfaces → Planar.



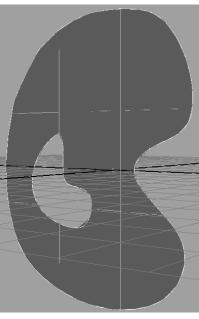


Figure 5.8

A planar surface based on a single curve (left); a planar surface based on a curve within a curve to create the cutout (right)

A planar surface is great for flat lettering, for pieces of a marionette doll or paper cutout, or for capping the ends of a hollow extrusion. It's usually best to create the planar surface as a polygon mesh, a technique you'll see later in this chapter in the "Using NURBS Surfacing to Create Polygons" section.

If you try creating a planar surface using a curve and notice that Maya doesn't allow it, verify that the curve is perfectly flat. If any of the CVs aren't on the same plane as the others, the planar surface won't work.

Beveled Surface

With the Bevel Surface function you take an open or closed curve and extrude its outline to create a side surface. It creates a bevel on one or both corners of the resulting surface to create an edge that can be made smooth or sharp (see Figure 5.9). The many options in the Bevel tool allow you to control the size of the bevel and depth of extrusion, giving you great flexibility. When a bevel is created, you can easily cap the bevel with planar surfaces.

To create a bevel, draw and select your curve and then choose Surfaces → Bevel.

Maya also offers a Bevel Plus surface, which has more creation options for advanced bevels. A beveled surface is great for creating 3D lettering, for creating items such as bottle caps or buttons, and for rounding out an object's edges.

Boundary Surface

A boundary surface is so named because it's created within the boundaries of three or four surrounding curves. For example, you draw two vertical curves opposite each other to

define the two side edges of the surface. Then you draw two horizontal curves to define the upper and lower edges. These curves can have depth; they need not be flat for the boundary surface to work, unlike a planar surface. Although you can select the curves in any order, it's best to select them in opposing pairs. In Figure 5.10, four curves are created and arranged to form the edges of a surface. First, you would select the vertical pair of curves because they're opposing pairs; then, you would select the second two horizontal curves before choosing Surfaces → Boundary.

A boundary surface is useful for creating shapes such as car hoods, fenders, and other formed panels, especially when created as polygon mesh.

Figure 5.9 A curve before and after it's beveled. The beveled surface has been given a planar cap.

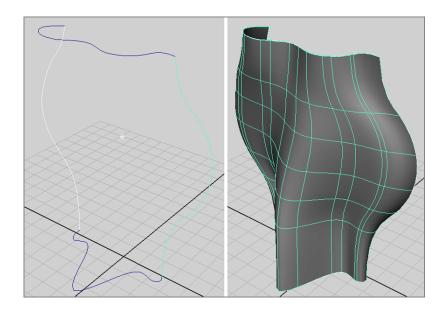


Figure 5.10 Four curves arranged to create the edges for a surface (left); the resulting boundary surface formed from the four curves (right)

Using NURBS Surfacing to Create Polygons

You can easily create swatches of polygon surfaces instead of NURBS surfaces by using any of these NURBS surfacing tools. To create a polygonal surface instead of a NURBS surface, open the option box for any particular NURBS tool.

Try This Draw two CV curves as you did at the beginning of this chapter, both with the same number of CVs. Choose Surfaces → Loft □ and click the Polygons option for Output Geometry. Set Type to Quads and leave the Tessellation method as Standard fit. Set the Chord height ratio to 0.95 and the 3D delta to 0.07, as shown in Figure 5.11. Click Apply.

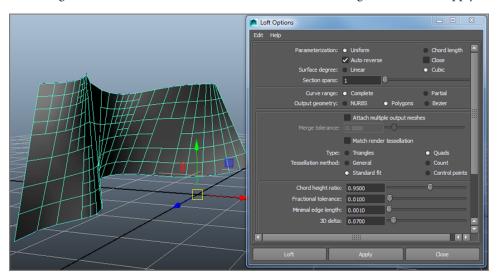


Figure 5.11 Create two curves

and loft between them to make a polygon mesh.

The creation options that appear at the bottom of the option window affect the tessellation of the resulting surface; that is, you use them to specify the level of detail and the number of faces with which the surface is created.

The Standard Fit Tessellation method uses the fewest faces to create the surface without compromising overall integrity. The sliders adjust the resulting number of faces in order to fit the finer curvature of the input curves, particularly Chord Height Ratio and 3D delta.

The Chord Height Ratio determines the amount of curve in a particular region and calculates how many more faces to use to give an adequate representation of that curved area with polygons. Values approaching 1.0 create more faces and better detail with surfaces that have multiple or very intense curves. Lower 3D Delta values will fit smaller faces around tight curves and lower the Fractional Tolerance settings, giving you a smoother surface and a greater number of faces.

Construction History

History has to do with how objects react to change. Leaving History on when creating objects allows you to adjust the original parameters that created that object. For example, the loft you just created will update whenever the curves you used to create the loft move or change shape.

Try This Select the new poly surface you just lofted in the previous "Try This" and open the Attribute Editor. Select the nurbs Tessellate1 tab and open the Advanced Tessellation Options heading. In the Standard Fit Options heading, experiment with changing the Chord Height Ratio and Delta values to see how the poly surface re-creates itself.

Also try selecting the CVs of the original curves (RMB+click one of the two curves you made and select Control Vertex from the marking menu). Adjust the curve, and the surface will re-create itself because of surface history.

You must toggle History on before you create the objects if you want History to be on for the objects.

In the status line, the History icon () toggles History on and off. Why wouldn't you want History turned on for everything? After a long day of modeling, having History on for every single object can slow down your scene file, adding unnecessary bloat to your workflow. But it isn't typically a problem on most surface types unless the scene is huge, so you should leave it on while you're still modeling.

If you no longer want a surface or an object to retain its history, you can selectively delete it from the surface. Select the surface, and choose Edit → Delete By Type → History. You can also rid the entire scene of History by choosing Edit → Delete All By Type → History. Just don't get them mixed up!

Other Tessellation Methods

Standard Fit will probably do the best job in most situations; however, here are the other tessellation methods you can use:

- The General Tessellation method creates a specific number of lines, evenly dividing the horizontal (U) and vertical (V) into rows of polygon faces.
- The Control Points method tessellates the surface according to the number of points on the input curves. As the numbers of CVs and spans on the curves increase, so does the number of divisions of polygons.
- The Count method simply relies on how many faces you tell it to make—the higher the count, the higher the tessellation on the surface. Experiment with the options to get the best poly surface results.

Converting a NURBS Model to Polygons

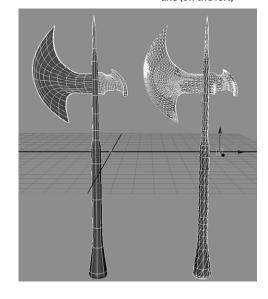
Some people prefer to model on NURBS curves and either create poly surfaces or convert to polygons after the entire model is done with NURBS surfaces. Ultimately, you'll find your own workflow preference, but it helps greatly if you're comfortable using all surfacing methods. In the following section, you'll convert a NURBS model to polygons.

Try This Convert a NURBS-modeled axe into a poly model like one that might be needed in a game.

Open axe_model_v1.mb in the Scenes folder of the Axe project from the companion web page. The toughest part of this simple process is getting the poly model to follow all the curves in the axe with fidelity, so you'll have to convert parts of the axe differently. Follow these steps:

- 1. Grab the handle, and choose Modify → Convert → NURBS To Polygons □. Use the default presets. If need be, reestablish your settings by choosing Edit → Reset Settings; the handle converts well to polygons. Click Apply, and a poly version of the axe handle appears on top of the NURBS version. Move it eight units to the right to get it out of the way. You'll move the other parts eight units as well to assemble the poly axe properly.
- 2. Select the back part of the axe head. All those surfaces are grouped together to make selection easy. The default settings will work for this part as well, so click Apply and move the resulting model eight units to the left.
- 3. The front of the axe head holds a lot of different arcs, so you'll have to create it with finer controls. Change Fractional Tolerance from 0.01 to 0.0005. This yields more polygons but finer-curved surfaces. Figure 5.12 shows the result.

Figure 5.12 A faithful high-poly conversion (on the right) of a NURBS axe (on the left)



Using Artisan to Sculpt NURBS

Imagine that you can create a NURBS surface and sculpt it using your cursor the way hands mold the surface of wet clay. As you saw with the Sculpt Geometry tool in Chapter 4, you can push and pull vertices easily by virtually painting on the surface. You can sculpt NURBS surfaces much the same way by accessing Edit NURBS → Sculpt Geometry Tool □.

If you plan to sculpt a more detailed surface, be sure to create the surface with plenty of surface spans and sections. You'll try this tool a little bit in the next section.

Creating a Pair of Glass Candle Holders

Let's put some of this information to good use and build a pair of glass candle jars. You will use these glass jars in Chapters 7, 10, and 11 to create shaders (materials), light, and render them. Copy the project folder candleHolder from the book's companion web page, www.sybex.com/go/introducingmaya2015, and set your project to it.

Creating the Objects

You'll start with a profile curve for a Revolve surface:

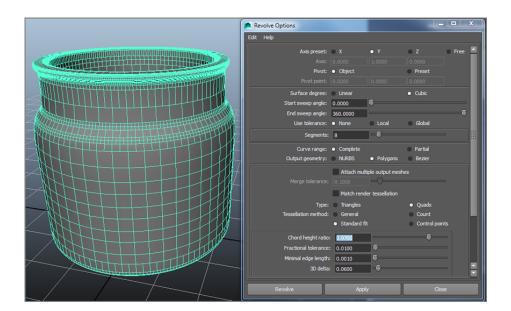
- 1. Click Create → CV Curve tool and turn on grid snaps () in the status bar.
- 2. In the Front view panel, click the origin to snap the first CV to (0,0,0). Snap the second and third CVs to the grid points on the left. Click the fourth CV slightly higher than the third CV to create a sharp curve. Figure 5.13 marks the first four CVs.

First four CVs

- Continue laying down CVs to match the shape shown in Figure 5.13. This will be the profile curve for the body of the jar.
- 3. In the Surfaces menu set, select Surfaces → Revolve \square . Make sure the Axis preset is Y and set the Output geometry to Polygons. Set Type to Quads. To get smooth curves in the mesh, increase Chord height ratio to 0.97 and lower 3D delta to **0.06**. Click Revolve to make the jar's body (Figure 5.14).
- 4. Now for the lid. In the Front view panel, click the Show menu in the view panel's menu bar and toggle off Polygons, as shown in Figure 5.15. This will disable polygon display in the Front view panel to make it easy to work on the next profile curve for the lid.

Figure 5.13

Starting with the first four CVs and then going clockwise around, complete your curve to match this shape.



5. The lid will have two parts: the main glass lid, and a plastic stopper that goes around the bottom of the glass part where it inserts into the jar. Create a curve to match Figure 5.16. Notice where the first few CVs are created and follow counterclockwise to lay down all the CVs as shown.

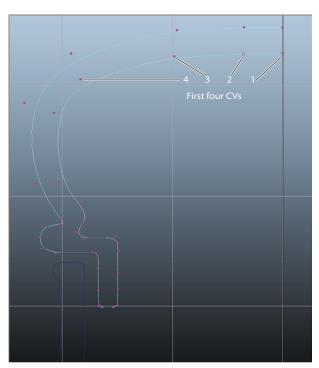


Figure 5.14

Revolve the surface with these settings.



Figure 5.15

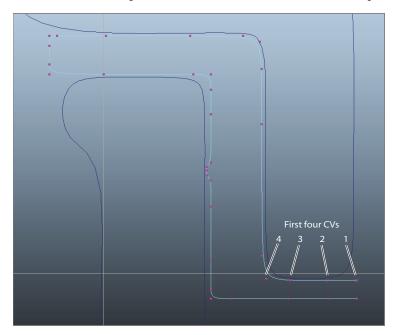
Turn off polygons in the Front view panel.

Figure 5.16

Start with the first CVs shown and follow the line counterclockwise to create this profile for the glass lid.

6. Now let's make a profile curve for the plastic sealer that goes around the base of the lid. In the Front view panel, create another curve to match the shape in Figure 5.17.

Figure 5.17 Profile curve for the plastic sealer for the lid



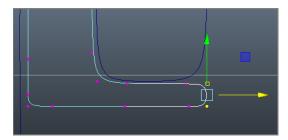


Figure 5.18 Close the curve and adjust the CVs to reshape the profile for the plastic stopper.

- 7. Now, the plastic sealer is open ended, so you need to close that curve to get a solid geometry when you revolve it. In Object Mode, select the curve and choose Edit Curves → Open/Close Curves. The end will close, but it will elongate. Select the end CVs and move them back to line up with the glass lid, as shown in Figure 5.18.
- 8. Now for the fun part! Go into the persp view panel, select the plastic sealer's profile curve, and choose Surfaces → Revolve just as you did last time, with the same settings originally shown in step 3 and Figure 5.14. Figure 5.19 (left) shows the plastic sealer.
- 9. What's a candle jar without a candle? You need the candle in the jar to form right up against the glass, so you'll use part of the same profile curve you used for the jar. Select the original jar body profile curve, and in the Outliner, change its name to glassProfile. With it still selected, choose Edit → Duplicate (or press Ctrl+D to copy the curve. Rename that curve from glassProfile to candleProfile.

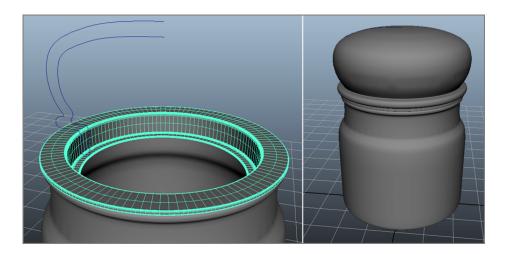


Figure 5.19 The plastic sealer (left) and the completed jar (right)

- 10. Select the glassProfile curve again and press Ctrl+H to hide it from view. Its entry in the Outliner will turn gray. RMB+click the duplicated curve (candleProfile) and select Curve Point from the marking menu. Click the curve and drag the little red point that shows up on the curve up the inside wall to right below where it begins to curve in, as shown in Figure 5.20. Release the mouse, and the point turns yellow. You are going to break the curve at this point.
- 11. Choose Edit Curves → Detach Curves to cut the curve at that point. The top of the curve should be green now and the bottom white. Select just the top part of the curve and press Delete to delete it. Figure 5.21 (left) shows the remaining curve you'll use to create the candle.
- 12. Draw a straight line curve as shown in the Front view panel to match Figure 5.21 (right). It's okay if the first CV isn't exactly on the candle's profile curve; close is good enough.
- 13. Now select both the straight line curve and the candle Profile curve and choose Edit Curves → Attach Curves □. Set Attach Method to Connect, set Multiple Knots to Remove, and uncheck Keep Originals, as shown in Figure 5.22. Click Attach, and the two curves will connect as shown to make the candle's profile curve.
- 14. With that candle curve selected, choose Surfaces → Revolve with the same settings as in step 3 to create the candle.

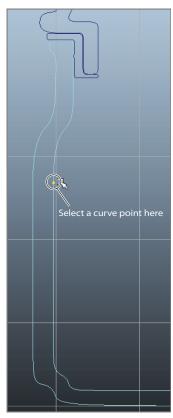
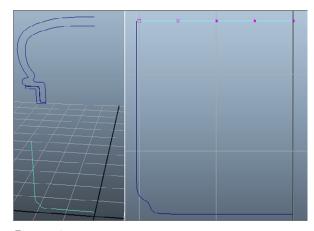
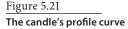


Figure 5.20 Selecting a point on curve





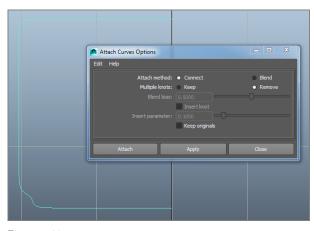


Figure 5.22 Attach the curves to complete the candle's profile.

You can edit the shape of any of these curves you've created, and because of History, the surfaces you created with Revolve will adjust to match the new profiles. To turn the glass profile curve (glassProfile) back on, select it in the Outliner and press Shift+H. You can easily edit the shape of the object to your liking now. Once you are done, simply delete the profile curves if you don't need them anymore.

Once you delete the profile curves, name the objects. Group the lid objects together (name it jarLid) and group the glass jar and candle together (name it jarCandle), as shown in Figure 5.23. Finally, in the Front view panel, select Show → Polygons (in the view panel's menu bar) to turn on the display of polygons in that view panel.

The candle jar can be found in candleModel_v01.mb in the scenes folder of the candle-Holder project. This file still retains the original profile curves for you. candleModel_v02.mb has the completed jar without the original profile curves and with the objects properly named and grouped.

A curve point is a point directly on a curve. You cannot move a curve point as you can a CV to adjust the shape of the curve. Curve points are used mostly for attach and detach functions.



Figure 5.23 The final candle's hierarchy

Detailing the Objects

Now let's add a little detail to the candle inside the jar using the Sculpt Geometry tool. Continue with your own file, or open candleModel_v02.mb in the scenes folder of the candleHolder project.

1. Select the outer glass jar and press Ctrl+H to hide it. In the Outliner, select the jarLid group and press Ctrl+H to hide the lid and plastic sealer, too.

2. Switch to the Polygons menu set. Choose Mesh Tools → EDIT → Sculpt Geometry Tool \square . It's especially important to open the Options Box for this tool.

Your cursor changes to the Artisan brush, as shown in Figure 5.24. The red circle around the brush cursor and the lettering display the type of brush you're currently using. When the red lines point outside the circle and the lettering reads Ps, you're using the push brush that pushes in the surface as you paint it. The black arrow pointing toward the sphere's center is a measurement of the Max Displacement slider in the tool settings. This sets how far each stroke pushes in the surface. The lower the number, the less the brush affects the surface.

- 3. If you click the candle, it will push in the geometry quite a bit. Undo anything you may have done with this tool. In the Options, set Radius(U) to 0.3 to make the brush smaller. Then set the Max displacement to 0.1 to limit how much you deform the mesh. Next, set Opacity to **0.1**. This ensures that the painting is layered slowly; each successive brush stroke will push the geometry slowly further. Paint the top of the candle slightly to create an uneven surface. Figure 5.24 shows the settings and the resulting uneven candle top.
- 4. In the Options, click the Pull operation (). Slowly raise the middle top of the candle to create a place for the wick, as shown in Figure 5.25.

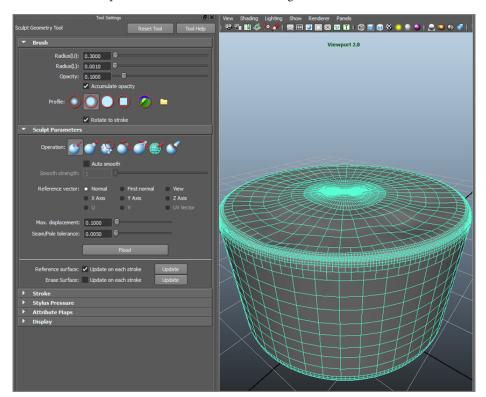


Figure 5.24

The Sculpt Geometry tool lets you mold your surface by painting on it. Here, the brush is set to push in the surface of the sphere as you paint.

- 5. Press W to exit the Sculpt Geometry tool and then close the Tool Options. With Interactive Creation turned off, create a polygon cylinder with Height divisions of 8 and Cap divisions of 4. Scale and place the cylinder to be the wick of the candle, as shown in Figure 5.26 (left).
- 6. Select vertices and move and rotate them around to give the cylinder a more interesting shape. Then use the Sculpt Geometry tool to misshape the cylinder as if the wick has already been burned, as seen in Figure 5.26 (right). Press 3 to see a smooth preview of the wick as you work.

Figure 5.25 Raise the top center a little for a place to put the candle's wick.

Figure 5.26 Place a cylinder as the candle's wick (left) and use the **Sculpt Geometry** tool and manual vertex manipulation to shape it into an already burned wick as shown (right).

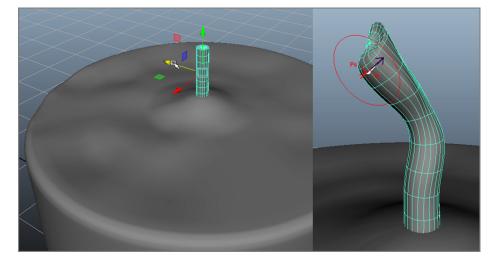




Figure 5.27 Group the wick into the hierarchy.

Additionally, Smooth () blends the pushed-in and pulled-out areas of the surface to yield a smoother result. Erase (simply erases the deformations on the surface, setting it back to the way it was before. In general, if you plan to sculpt a detailed surface, be sure to create the mesh with plenty of faces.

- 7. Once you are happy with the wick, smooth it by choosing Mesh → SHAPE → Smooth □. Set the Division levels to 1 and click Smooth.
- 8. Select the wick and erase the history on it by choosing Edit → Delete By Type → History. Name the object wick and place it into the jarCandle group, as shown in Figure 5.27.

9. In the Outliner, select the jarLid group and press Shift+H to unhide it. Also select the glassJar object (under the jarCandle group) to unhide that as well. Then group both jarLid and jarCandle groups together into a new group called candleGroup.

You can check your work against the file candleModel_v02.mb in the scenes folder of the candleHolder project. With the glass candle jar holder complete, you'll be able to use it later to create a glass and candle wax shader in Chapter 7 and then render the object with reflections and refractions in Chapter 11. Nice job! Go grab a cookie.

Modeling with Simple Deformers

In many ways, deformers are the Swiss Army knives of Maya animation, except that you can't open a bottle with them. Deformers are handy for creating and editing modeled shapes in Maya. These tools allow you to change the shape of an object easily. Rather than using CVs or vertices to distort or bend an object manually, you can use a deformer to affect the entire object. Popular deformers, such as Bend and Flare, can be powerful tools for adjusting your models quickly and evenly, as you're about to see.

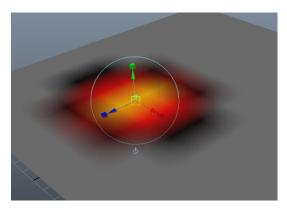
Nonlinear deformers, such as Bend and Flare, create simple shape adjustments for the attached geometry, such as bending the object. You can also use deformers in animation to create effects or deformations in your objects. You'll explore this later in the book.

Soft Modification Deformer

First, let's try using the Soft Modification tool. Choose Modify → Transformation Tools → Soft Modification Tool, and its icon (🚵) appears below the Tool Box. This tool allows you to select an area on a surface or model and make any adjustments in an interesting way. These adjustments taper off away from the initial place of selection, giving you an easy way to soft-modify an area of a model, much like lifting up a tablecloth from the middle.

To try the Soft Modification tool, follow these steps:

- 1. In a new scene create a Polygon plane by choosing Create → Polygon Primitives → Plane □. Set both the Width Divisions and Height Divisions sliders to 10, and click Create.
- 2. Click and drag a plane on the grid if Interactive Creation is turned on; otherwise, a plane appears at the origin on your grid. Select the Scale tool, and scale the plane up to about the size of the grid.
- 3. Select Modify → Transformation Tools → Soft Modification Tool and click the plane somewhere just off the middle. Doing so creates an S and a special manipulator to allow you to move, rotate, or scale this soft selection (see Figure 5.28). You also see a yellow-to-red-to-black gradient around the S manipulator. This shows you the area and degree of influence, where yellow moves the most and black the least.
- 4. Grab the cone handle and drag it up to move the soft selection up. Notice that the plane lifts up in that area only, gradually falling off.





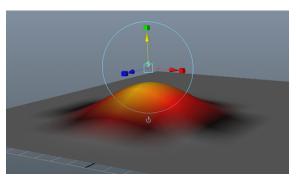


Figure 5.29 Lifting an area of the Polygon plane

5. Grabbing the cube handle scales the soft selection, and dragging on the circle rotates it. After you've finished making your soft adjustments, make sure to press W for the Move tool and to exit the Soft Modification tool. Otherwise, you will create new soft modification handles every time you click the mesh.

You can go back to any soft selection by selecting the *S* on the surface for later editing. You can place as many soft selections as you need on a surface. Figure 5.29 shows the soft modification adjusting the plane.

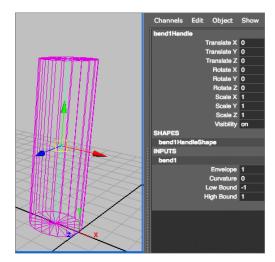
Modeling Using the Bend Deformer

Let's apply a deformer. In a new Maya scene, you'll create a polygonal cylinder and bend it to get a quick idea of how deformers work. Follow these steps:

1. Choose Create → Polygon Primitives and turn on Interactive Creation. Then, choose Create → Polygon Primitives → Cylinder. Click and drag to create the base. Make it a few units in diameter; the exact sizing isn't important here. Click and drag to make the height of the cylinder 7 or 8 units. Make sure your create options are set to the defaults so that they're consistent with these directions.

To make sure the settings are at their defaults, open the command's option box and click the Reset Settings or Reset Tool button.

2. With the cylinder selected, switch to the Animation menu set by pressing F2 and choosing Create Deformers → Nonlinear → Bend. Your cylinder turns magenta if you are in wireframe display mode, and a thin line appears at the center of the cylinder, running lengthwise. Figure 5.30 shows the deformer and its Channel Box attributes. Depending on your settings, your deformer may be created in a different axis than the one pictured. You can reset the deformer's options as needed. Click bend1 in the Channel Box to expand the deformer's attributes as shown in the figure.



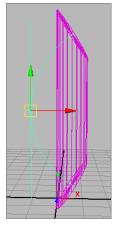




Figure 5.30 Creating the Bend deformer

Figure 5.31 Notice the problem with this cylinder?

Figure 5.32 Increase Subdivisions Height to 12.

- 3. Click Curvature, and enter a value of **60**. Notice that the cylinder takes on an odd shape, as shown in Figure 5.31. The Bend deformer itself is bending nicely, but the geometry isn't. Also, the geometry is offset and weird-looking because there aren't enough divisions in the geometry to allow for a proper bend.
- 4. Select the cylinder, and click polyCylinder1 in the Channel Box to expand the shape node's attributes. Enter a value of 12 for the Subdivisions Height attribute (see Figure 5.32), and your cylinder will bend with the deformer properly, as shown in Figure 5.33.
- 5. Try adjusting the Bend deformer's Low Bound and High Bound attributes. This allows you to bend one part of the cylinder without affecting the other. For example, set the Bend deformer's High Bound option to 0.25 instead of 1. This causes the top half of the cylinder to bend only one quarter of the way up and continue straight from there.

Experiment with moving the Bend deformer, and see how doing so affects the geometry of the cylinder. The deformer's position plays an important role in how it shapes an object's geometry.

Adjusting an Existing Axe Model

In this exercise, you'll take a NURBS model of an axe and fine-tune the back end of the axe head. In the existing model, the back end of the axe head is blunt, as you can see in Figure 5.34. You'll need to sharpen the blunt end with a nonlinear deformer. Open the AxeHead_v01.ma file in the scenes folder of the Axe project from the companion web page, and follow these steps:

1. Select the top group of the axe head's back end. To do so, open the Outliner, and select axeHead_Back (see Figure 5.35).

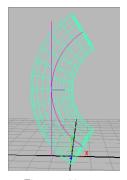
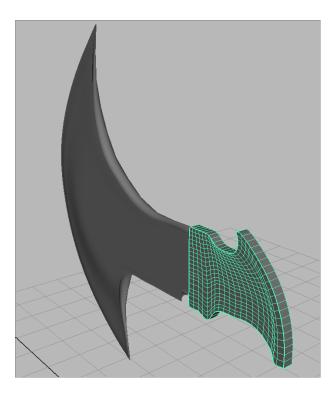


Figure 5.33 The cylinder bends properly now that it has the right number of divisions.

Figure 5.34

The axe head is blunt.



- 2. Press F2 to switch to the Animation menu set.
- 3. Create a Flare deformer by choosing Create Deformers → Nonlinear → Flare. The Flare deformer appears as a cylindrical object (see Figure 5.36).
- 4. Rotate the deformer 90 degrees in the *Z*-axis, as shown in Figure 5.37.

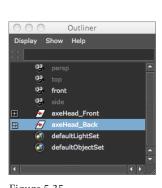
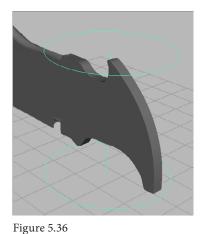


Figure 5.35

Select the back of the axe head.



The Flare deformer appears as a cylinder.

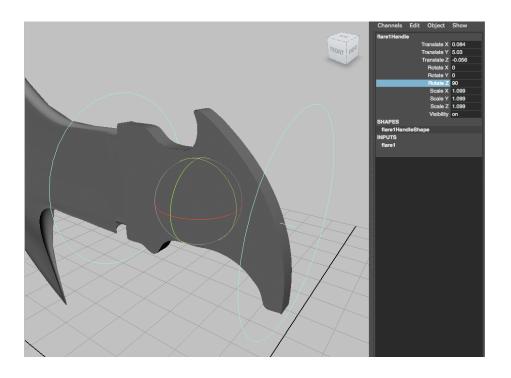


Figure 5.37 Rotate the deformer 90 degrees in the Z-axis.

5. Open the Attribute Editor (Ctrl+A), click the flare1 tab to access the Flare controls, and enter the following values:

ATTRIBUTE	VALUE
Start Flare Z	0.020
High Bound	0.50

These values taper in the end of that part of the axe head, as shown in Figure 5.38. This is a much easier way of sharpening the blunt end than adjusting the individual

CVs of the NURBS surfaces.

Deformers use History to distort the geometry to which they're attached. You can animate any of the attributes that control the deformer shapes, but in this case you're using the deformer as a means to adjust a model. When you get the desired shape, you can discard the deformer. However, simply selecting and deleting the deformer will reset the geometry to its original blunt shape. You need to pick the axeHead_Back geometry group (not the deformer) and delete its History by choosing Edit → Delete By Type → History.

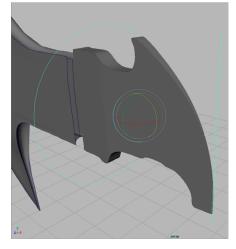


Figure 5.38 Sharpen the axe's back edge.

The Lattice Deformer

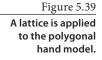
When a model requires more intricate editing with a deformer, you'll need to use a lattice. A lattice is a scaffold that fits around your geometry. The lattice object controls the shape of the geometry. When a lattice point is moved, the lattice smoothly deforms the underlying geometry. The more lattice points, the greater control you have. The more divisions the geometry has, the more smoothly the geometry will deform.

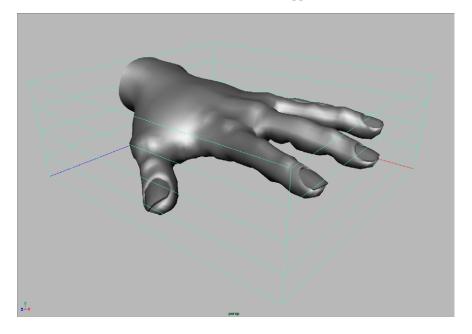
Lattices are especially useful when you need to edit a relatively complex poly mesh or NURBS surface that is too dense to edit efficiently directly with CVs or vertices. With a lattice, you don't have to move the individual surface points.

Lattices can work on any surface type, and a single lattice can affect multiple surfaces simultaneously. You can also move an object through a lattice (or vice versa) to animate a deformation effect, such as a golf ball sliding through a garden hose.

Creating an Alien Hand

Make sure you're in the Animation menu set. To adjust an existing model or surface, select the model(s) or applicable groups to deform, and choose Create Deformers → Lattice. Figure 5.39 shows a polygonal hand model with a default lattice applied. The top node of the hand has been selected and the lattice applied.





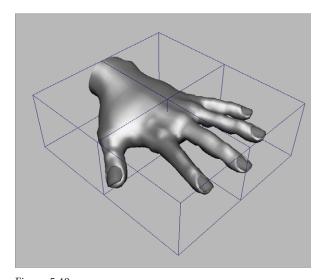
Your objective is to create an alien hand by thinning and elongating the hand and each of the fingers—we all know aliens have long, gawky fingers! Because it would take a lot of time and effort to achieve this by moving the vertices of the poly mesh itself, using lattices here is ideal.

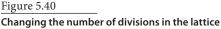
To elongate and thin the entire hand, load the scene file detailed_poly_hand.ma from the Poly_Hand project from the companion web page, and follow these steps:

- 1. Select the top node of the hand (poly_hand) in the Outliner, and choose Create Deformers → Lattice. Doing so creates a default lattice that affects the entire hand, fingernails and all. Although you can change the lattice settings in the option box upon creation, you'll edit the lattice after it's applied to the hand.
- 2. The lattice is selected after it's created. Open the Attribute Editor, and click the ffd1LatticeShape tab. The three attributes of interest here are S Divisions, T Divisions, and U Divisions. These sliders control how many divisions the lattice uses to deform its geometry. Set S Divisions to 3, T Divisions to 2, and U Divisions to 3 for the result shown in Figure 5.40.

A $3\times2\times3$ lattice refers to the number of division lines in the lattice as opposed to the number of sections; otherwise, this would be a 2×1×2 lattice!

- 3. With the lattice selected, press F8 to switch to Component mode and display the lattice's points. You'll use these points to change the overall shape of the hand. Select the lattice points on the thumb side of the hand, and move them to squeeze in that half of the hand. Notice how only that zone of the model is affected by that part of the lattice.
- 4. Toggle back to Object mode (press F8 again) and scale the entire lattice to be thinner in the *Z*-axis and longer in the *X*-axis. The entire hand is deformed in accordance with how the lattice is scaled (see Figure 5.41).





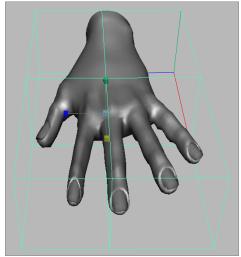


Figure 5.41 Lengthen the hand using the deformer.

Now that you've altered the hand, you have no need for this lattice. If you delete the lattice, the hand will snap back to its original shape. You need to delete the construction history on the hand to get rid of the lattice, as you did with the axe head exercise earlier in this chapter.

5. Select the top node of the hand and choose Edit → Delete By Type → History.

Creating Alien Fingers

The next step is to elongate the individual fingers and widen the knuckles. Let's begin with the index finger. Follow these steps:

1. Select the top node of the hand and create a new lattice as before. It forms around the entire hand.

Although you can divide the lattice so that its divisions line up with the fingers, it's much easier and more interactive to scale and position the entire lattice so it fits around the index finger only.

Simply moving and scaling the selected lattice will deform the hand geometry. You don't want to do this. Instead, you need to select the lattice and its base node. This lets you change the lattice without affecting the hand.

- 2. In the Outliner, select both the ffd1Lattice and ffd1Base nodes (see Figure 5.42).
- 3. Scale, rotate, and transform the lattice to fit around the index finger, as shown in Figure 5.43.

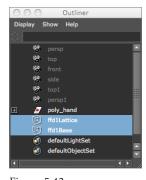
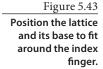
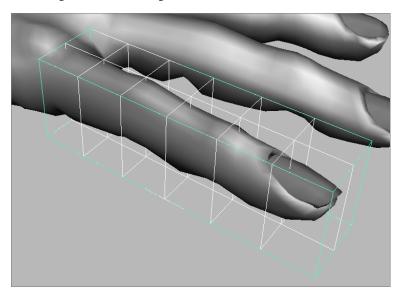


Figure 5.42 Select the lattice and base nodes.





- 4. Deselect the base, and set the lattice S Divisions to 7, T Divisions to 2, and U Divisions to 3.
- 5. Adjust the lattice to lengthen the finger by pulling the lattice points (see Figure 5.44). Pick the lattice points around each of the knuckles individually and scale them sideways to widen them.

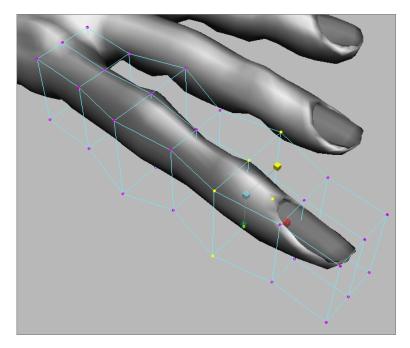


Figure 5.44 Flare out the knuckles.

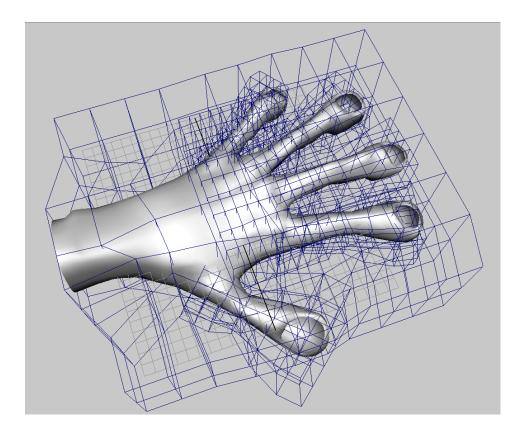
- 6. To delete the lattice and keep the changes to the finger, select the top node of the hand and delete its History.
- 7. Repeat this entire procedure for the rest of the fingers to finish your alien hand. (Try to creep out your younger sister with it.)
- 8. To get rid of all the lattices and keep the alien shape, select the hand meshes and choose Edit > Delete by Type > History.

The alien hand in Figure 5.45 was created by adjusting the polygonal hand from this exercise using only lattices.

As you can see, lattices give you powerful editing capabilities without the complication of dealing with surface points directly. Lattices can help you reshape an entire complex model quickly or adjust minor details on parts of a larger whole.

In Chapter 8, "Introduction to Animation," you'll animate an object using another type of deformer. You'll also learn how to deform an object along a path.

Figure 5.45 The human hand model is transformed into an alien hand by using lattices to deform the geometry.



Animating Through a Lattice

Lattices don't only work on polygons; they can be used on any geometry in Maya and at any stage in your workflow to create or adjust models. You can also use lattices to create animated effects. For example, you can create the effect of a balloon squeezing through a pipe by animating the balloon geometry through a lattice.

In the following exercise, you'll create a NURBS sphere with 8 sections and 16 spans and an open-ended NURBS cylinder that has no end caps:

- 1. Choose Create → NURBS Primitives → Sphere □, set Number of Sections to 8 and Number of Spans to 16, and create the sphere.
- 2. Choose Create → NURBS Primitives → Cylinder □, and check None for the Caps option. Scale and arrange the sphere balloon and cylinder pipe as shown in Figure 5.46.

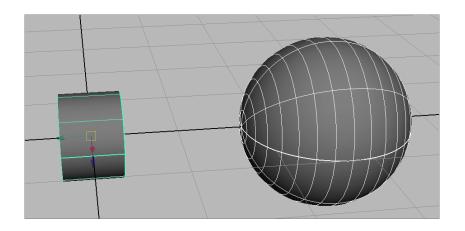


Figure 5.46 Arrange the balloon and pipe.

3. Select the balloon and create a lattice for it (see Figure 5.47). (From the Animation menu set, choose Create Deformers - Lattice.) Set the S, T, and U Divisions to 4, 19, and 4, respectively. You set this number of lattice divisions to create a smoother deformation when the sphere goes through the pipe.

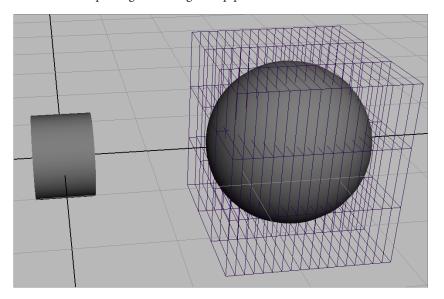
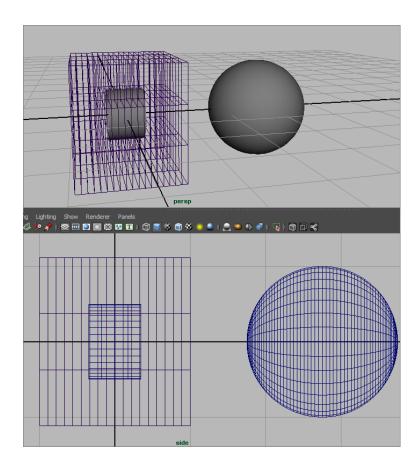


Figure 5.47 Create a lattice for the sphere.

4. Select the lattice and its base in the Outliner (ffd1Lattice and ffd1Base nodes), and move the middle of the lattice so it fits over the length of the pipe (see Figure 5.48).

Figure 5.48

Relocate the lattice to the cylinder.



- 5. Deselect the lattice base and choose Component mode for the lattice. Select the appropriate points and shape the lattice so the middle of the lattice fits into the cylinder (see Figure 5.49).
- 6. Select the sphere and move it back and forth through the pipe and lattice. Notice how it squeezes to fit through. If you look closely, you'll see that the sphere starts to squeeze a little before it enters the pipe. You'll also see parts of the sphere sticking out of the very ends of the pipe. This effect, in which geometry passes through itself or another surface, is called *interpenetration*. You can avoid this by using a more highly segmented sphere and lattice. If you try this exercise with a lower-segmented sphere and/or lattice, you'll notice the interpenetrations even more. Figure 5.50 shows the balloon squeezing through the pipe.

In a similar fashion, you can create a lattice along a curve path and have an object travel through it. You'll try this in Chapter 8.

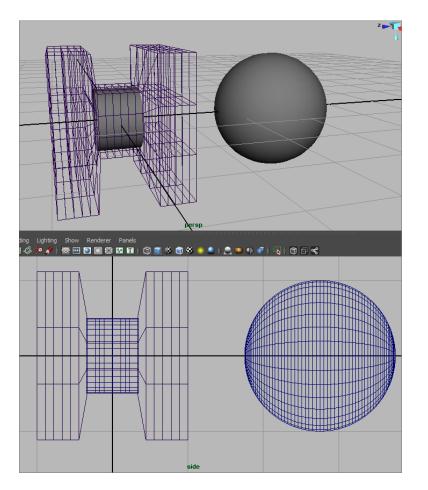


Figure 5.49 Squeeze in the lattice points to fit the cylinder.

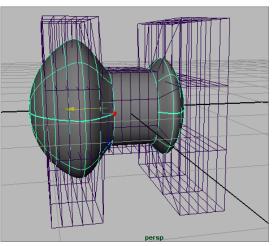


Figure 5.50 Squeezing the balloon through the pipe using a lattice deformer

Summary

In this chapter, you tackled NURBS modeling by going through the usual surfacing tools, from lofting to revolving. Then you explored the implications of surface History and how surfaces adjust to changes when History is enabled. You then put those lessons to work on creating a pair of glass candle holders.

This chapter also covered various modeling techniques to help you break away from typical ways of thinking. You learned how to use a lattice to adjust a polygon hand model into an alien hand, as well as how to animate a balloon pushing through a pipe. Different workflows give you the flexibility to choose your own modeling style. To make good choices, however, you'll need to practice.

Keep at it; model everything you can get your hands and eyes on. As you're doing that, stay on top of how you organize your nodes and keep everything named and organized.

For further practice, use this chapter as a reference to create some of the following models using NURBS surfaces and lattices to aid in shaping polygons:

Bathroom Sink Snap a few digital stills of your bathroom sink, or find some pictures on the Internet. A sink will give you a great chance to explore NURBS surfacing, making pristine curves and smooth surfaces. It may be a bit involved, but it's not overwhelming.

Cartoon Head Use Maya Artisan and the Sculpt Geometry tool to turn an ordinary sphere into a cartoonish head. It's fun to use Sculpt Geometry to model. Try to make the head using only Artisan.

Computer Mouse A PC or Mac mouse makes a great simple NURBS model.



Practical Experience!

It's $time\ to\ put$ what you've learned so far into action. In this chapter, you'll build a child's airplane toy. The plane project uses poly and NURBS modeling techniques to give you some practical experience with a larger project in the Autodesk® Maya® software.

Learning Outcomes: In this chapter, you will be able to

- Use bevels and extrusions efficiently
- Manipulate curves to create poly meshes with Revolves and Loft Surfaces
- Create a shape with a path extrusion
- See the differences between the Maya and Modeling Toolkit workflows
- Use reference images to shape your models
- See how to set up views with grid lines to align reference images for modeling
- See how images are lined up to make reference images for modeling
- Work in the Hypershade to assign image maps to objects in the scene
- Use a Boolean operation to cut holes in a mesh

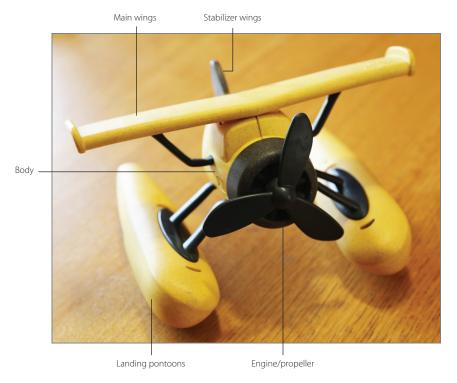
Evaluating the Toy Plane

Download the entire Plane project folder structure from the book's companion web page (www.sybex.com/go/introducingmaya2015) to your computer's hard drive and set your project to that location.

Also on the book's website (as well as the author's site www.koosh3d.com) are video tutorials taking you through parts of this exercise.

Figure 6.1 shows a bathtub toy airplane that you'll be modeling first in this chapter. There's certainly enough detail in this object to make it a good exercise, but it won't be difficult to complete. You can always return to this exercise to add more of your own detail or even redesign it for more challenge, which is something I highly recommend.

Figure 6.1 The toy plane is cute!



Study the photos (Figures 6.1, 6.2, and 6.3) carefully to get an understanding of the components that make up this object. You will model the parts individually instead of attacking the entire shape as a single object. This way of thinking helps complex objects become easier to understand and model. The plane consists of the body, landing pontoons, engine/propeller, stabilizer wings, and main wings, as shown in Figure 6.1. In Chapter 3's decorative box exercise, you used reference planes to give you an accurate guide to build the box, and you'll do the same with the individual parts of the plane.

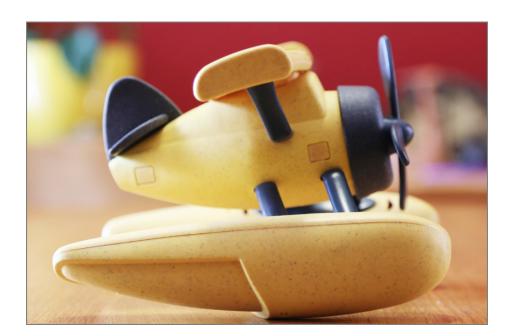


Figure 6.2 The side photo of the plane



Figure 6.3 The top photo of the plane

Building the Landing Pontoons

Let's start with the landing pontoons. You need to model only one and then make a duplicate and mirror it for the other side.

Reference Images

First you need to assemble the reference planes to make the model easier to build. Figure 6.4 shows the top, side, and front views of a pontoon scaled and lined up in Adobe Photoshop. The individual photos have been copied and pasted into a larger image, and grid lines are used to line up the proportions of the pontoon. Keep in mind that when you take a photo, in most cases there will be perspective shift, or parallax, in the image. Because of that shift, the different views of the same object will never exactly line up. Once each individual photo is scaled and lined up like in Figure 6.4, you can output each image as its own file to bring into Maya for top, side, and front reference.

Figure 6.4 The reference photos of the pontoon are lined up.



Now, you'll import the images and create the model reference planes on which to work.

Creating Reference Planes for the Images

The reference images of the toy airplane have already been created for you. You can find them in the sourceimages folder of the Plane project on the companion website. They're shown in Table 6.1.

Table 6.1 Reference views and sizes

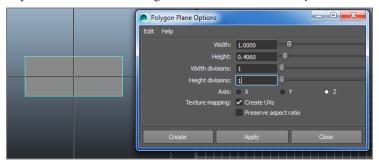
FILENAME	VIEW	IMAGE SIZE	ASPECT RATIO
pontoonTopReference.jpg	Тор	1600×650	1:0.406
pontoonSideReference.jpg	Side	1600×650	1:0.406
pontoonFrontReference.jpg	Front	1600×650	1:0.406

Why is the image resolution important? Well, it's not so much the resolution of the photos but rather the aspect ratio of each image. To properly map these images onto the planes you'll use in Maya, each plane has to be the same aspect ratio as its image. For example, an image that is 100×50 pixels has an aspect ratio of 2:1 and is, therefore, a wide horizontal rectangle. For it to map properly, the plane on which it's mapped in Maya must have a scale ratio of 2:1 so that it's also a wide horizontal rectangle. Otherwise, the image may distort, causing inaccuracies in your model. Since each of the pontoon references was made the same size, it's a little easier.

You'll need to create three planes for each of the three views. First make sure Interactive Creation is turned off; then follow these steps:

1. In a new scene, go to the front view panel and create a polygonal plane by choosing Create → Polygon Primitives → Plane □. This plane is for the front image, so in the

option box set Axis to **Z**, Width to **1**, and Height to 0.406. Set Width Divisions and Height Divisions both to 1. Make sure the check box for Preserve Aspect Ratio is deselected and Axis is set to **Z** to place the plane properly in the front view, as shown in Figure 6.5; click Apply. Name this plane **frontRefPlane**.



2. Switch to the side view panel. Create a second plane, again with a Width value of 1 and a Height value of **0.406**. Set Axis to **X**, and make sure the Preserve Aspect Ratio

check box remains unchecked. Name this plane **sideRefPlane**.

3. Switch to the top view panel. Create a third plane with Width remaining at 1, Height at **0.406**, but Axis now set to **Y**. Make sure the Preserve Aspect Ratio box is still unchecked. Your perspective panel should look like Figure 6.6. Name this plane topRefPlane.

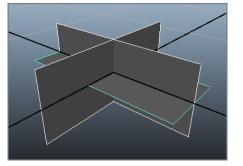


Figure 6.6 The three view planes are ready.

Figure 6.5

the top view

Creating a plane for

Mapping the Reference Planes

Now that all three image planes have been created with the proper aspect ratio, you're ready to map the photos to them to create the reference for the model.

1. Open the Hypershade (Window → Rendering Editors → Hypershade). Open a file browser in your OS and navigate to the Sourceimages folder of the Plane project on your hard drive.

Figure 6.7 Importing the photos into the Hypershade window

2. Click the pontoonSideReference.jpg file in your file browser and drag it into the Work Area of the Hypershade window to import the file image. Drag the other two images (pontoonFrontReference.jpg and pontoonTopReference.jpg) into the Hypershade one at a time, as shown in Figure 6.7.



If you don't see the image of the pontoon in the icons in the Hypershade like in Figure 6.7, simply right-click the file node icon and select Refresh Swatch from the marking menu.

- 3. You must create shaders for each of the planes. In the left panel of the Hypershade window, click the Lambert icon three times to create three new Lambert shaders, which should now display in the Work Area. For more information about shaders and texturing, see Chapter 7, "Autodesk" Maya Shading and Texturing." Remember, you can use Alt+MMB to move around the Work Area, and you can use Alt+RMB+click to zoom in and out.
- 4. While still in the Hypershade Work Area, hold down the Ctrl key and MMB+drag the side view's photo swatch onto the first Lambert shader icon. Maya automatically maps the image to the color of the Lambert shader, as shown in Figure 6.8.

As an alternate way to attach the image to the shader, double-click the Lambert and then MMB+drag the image to the Lambert shader's Color attribute in the Attribute Editor.

5. Ctrl+MMB+drag the other two images onto their respective Lambert shaders to connect them. Name the shaders lambertSide, lambertFront, and lambertTop, respectively, by double-clicking them to open the Attribute Editor and changing their names (see Figure 6.8).



Figure 6.8

Attach the images to the shaders and name them.

- 6. Assign the shaders to the reference planes. To do so, MMB+drag the Lambert shader that is connected to the side view image (lambertSide) to the side view plane in the perspective panel. Switch the perspective view into Texture display mode by pressing 6 while in that panel.
- 7. Drag the top view's Lambert shader to the top reference plane and drag the front Lambert shader to the front reference plane to assign the other two views. You should now have the three reference planes laid out as shown in Figure 6.9, although the top view plane needs to be re-oriented.
- 8. All isn't rosy yet! Select the top view plane in the persp view and rotate it 90 degrees in the Y-axis. Take a look around the reference planes, and you'll see that the front reference is slightly larger and a bit to the right of the top view reference plane. Select the front reference plane (frontRefPlane) and scale it to **0.925** in *X*, *Y*, and *Z*; set Translate X to -0.102 to center it properly (see Figure 6.10).

All three reference planes are mapped, though the top view is oriented wrong.

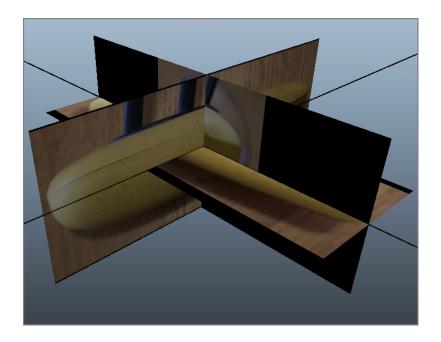
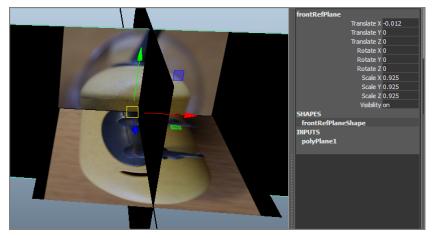


Figure 6.10

Scale and move the front reference pontoon to match the side reference pontoon. Done!



- 9. Remember that the scale of these planes is small right now. You don't need to use real-world units for this project. However, you should scale the reference planes so that you have a larger scale with which to work. Select all three reference planes and group them together by pressing Ctrl+G; name the group node pontoonRefPlanesGroup.
- 10. Select the top group node and uniformly scale it to 12 units, as shown in Figure 6.11.

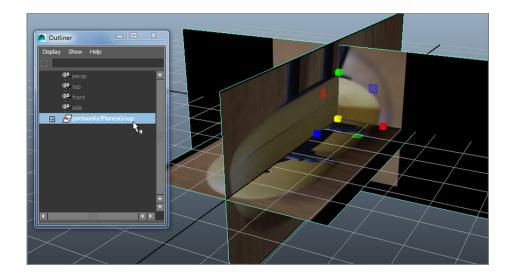


Figure 6.11 Group and then scale the reference planes up.

11. To give yourself more room to work, select the front view plane and move it back, as shown in Figure 6.12.

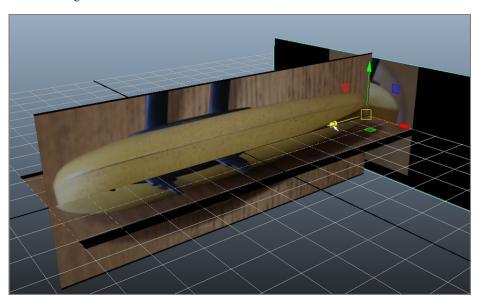


Figure 6.12 Move the front reference plane back.

12. Create a display layer for the planes to make it easy to manage them later. To do so, select pontoonRefPlanesGroup and, in the Layer Editor below the Channel Box, click the Create A New Layer And Assign Selected Objects icon (2).

Double-click the new layer and name it **pontoonRefLayer** in the window that pops up (see Figure 6.13). Your Layer Editor should resemble the one shown in Figure 6.14. For more on the Layer Editor, see Chapter 3.





Figure 6.13

Name the new layer.

Figure 6.14

The new layer in Layer Editor

To toggle the display of the reference planes to get them out of the way, you can simply toggle the box to the extreme left of the layer name, currently checked with a *V* for "visible" in Figure 6.14. Keep it visible for now.

Save your work. You can download the scene file pontoonModel_v01.ma from the Scenes folder of the Plane project from the companion website to check your work or skip to this point.

Just be sure to set your project to the Plane project on your hard drive after downloading the entire project from the website. Otherwise, the images for the reference planes may not show up.

To remain in step with this chapter, make sure Interactive Creation is turned off when you create any new primitives.

Blocking Out the Pontoon

The pontoon will start with a poly cube in the following steps:

1. In the top view, press 6 to see Textured mode, if you haven't already. In the Layer Editor, click the empty check box to the right of the Visibility toggle for the pontoonRefLayer layer twice so you see an R appear, as shown in Figure 6.15. This sets



that layer as a reference layer and makes the objects in the layer unselectable in the view panels. If you try clicking any of the three reference planes, nothing will happen, making it easier to model on top of.

Figure 6.15

Set the reference planes layer to Reference to make it unselectable in the view panels.

2. Create a poly cube. You need to see the cube as a wireframe but still see the reference image as a textured object. In the top view panel's menu (not the main Maya menu bar), select Shading → X-Ray (Figure 6.16) to allow you to see through the box but not lose your reference image display. Place and scale the cube over the pontoon; then RMB+click the mesh and select Vertex from the marking menu (to enter vertex mode) to move the corner vertices to roughly match the shape of the pontoon, as shown in Figure 6.17.

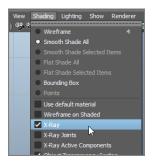


Figure 6.16

Set your view panel shading display to X-Ray.

- 3. Switch to the side view panel and enable X-Ray mode (Shading → X-Ray). Move the corner vertices to get a rough shape of the pontoon from the side, as shown in Figure 6.18.
- 4. Switch to Object mode (RMB+click the cube and select Object Mode from the marking menu, as shown in Figure 6.19). This is the fastest way to switch selection modes.
- 5. Make sure you are in the Polygons menu set and select the pontoon. Choose Mesh Tools → EDIT → Insert Edge Loop Tool. Insert nine edge loops along the pontoon, as shown in Figure 6.20 (top).



Figure 6.17 Shape a cube over the body of the

pontoon in the top view panel.

Figure 6.18 Shape the pontoon in the side view.

6. Using the marking menu, RMB+click the mesh to switch back to vertex selection. While in the side view panel, move the vertices of the new edge loops to better match the shape of the pontoon's side, as shown in Figure 6.20 (bottom). Don't worry about the dark gray arms that attach the pontoon to the body of the plane; shape your cube only to the yellow pontoon for now.

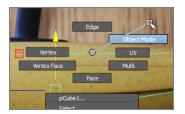


Figure 6.19

Select Object Mode in the marking menu. Insert nine edge loops along the pontoon's side (top) and then move vertices to better match the shape in the side view panel (bottom).

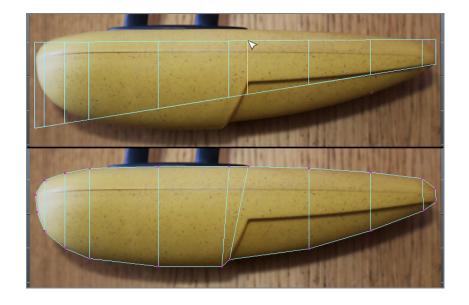




Figure 6.21

Move vertices in the top view panel to better match the pontoon's shape.

- 7. Switch to the top view panel and move the vertices to better match the shape from the top, as shown in Figure 6.21.
- 8. In the side view panel, insert three edge loops (Mesh Tools → EDIT → Insert Edge Loop Tool), as shown in Figure 6.22. Save your work as the next version to avoid writing over your older work.

Shaping the Pontoon

The scene file pontoonModel_v02.ma in the Scenes folder of the Plane project will bring you to this point. Now, let's shape the pontoon some more.

1. In the persp view, rotate your view (a.k.a. tumble) to see the bottom of the pontoon. Make sure X-Ray is off so you can see the pontoon as a solid object. In the Layer Editor, toggle the shading mode of the pontoonRefLayer by clicking the R toggle until it displays a *T*. The planes turn to a gray wireframe.

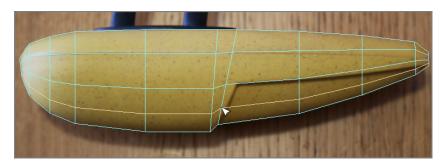
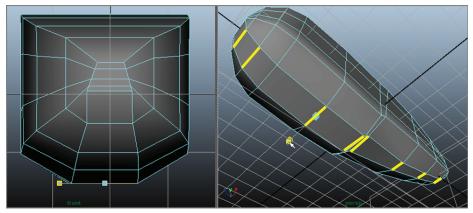


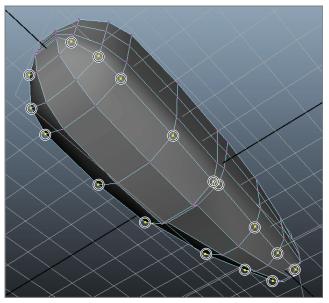
Figure 6.22 Insert three edge loops as shown in the side view panel.

2. Switch to edge selection mode. One by one from front to back, select the bottom row of edges on the pontoon's underside, and scale them in the *X*-axis *only* to taper the belly of the shape, as shown in Figure 6.23. You can shape it to your liking, without worrying about much accuracy to the photo references. You can refer to Figures 6.2 and 6.3 for a better look at the two pontoons' shape.



Taper these edges

3. In the persp view panel, select the vertices one at a time on the section up from the very bottom that you just adjusted in the previous step and shape them slightly to continue the taper of the belly to your liking. Figure 6.24 shows the pontoon so far.



Adjust these vertices

Figure 6.23

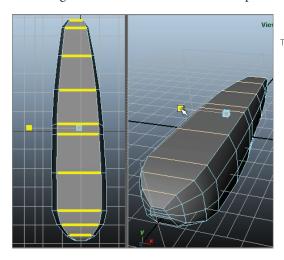
Shape the bottom of the pontoon to your fancy by scaling down the bottom row of edges one at a time in the X-axis.

Figure 6.24

Keep shaping the pontoon, now by moving the vertices that are one segment higher than the bottom edges you just adjusted.

Figure 6.25 Taper the top edges to your liking.

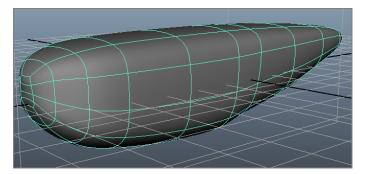
4. Now tumble your view in the persp view panel to the top of the pontoon. Select the top row of edges and scale them in a little bit to taper the top slightly, as shown in Figure 6.25.



Taper these edges

5. The shape of the pontoon can differ from the reference planes a bit; the overall proportion/shape was what you needed to get from the photos. You can continue to shape the pontoon to your liking. Once you are happy with it, select the pontoon and press 3 in the persp view panel to see a smooth preview. It looks much more like the pontoon when you see it in Smooth Preview (Figure 6.26). Save your work as a new version number!

Figure 6.26 The base pontoon



Attaching the Arms

The scene file pontoonModel_v03.ma in the Scenes folder of the Plane project will bring you to this point. Now let's tackle the gray arms that will eventually attach the pontoon to the plane's body. You'll also start using Modeling Toolkit in the following steps:

1. Click the Modeling Toolkit icon in the upper right of the Maya UI, as shown in Figure 6.27, to enable Modeling Toolkit. The right side of the UI now displays the

- suite of tools in the toolkit, and a new tab appears alongside the Channel Box/Layer Editor and Attribute Editor tabs.
- 2. In the top view panel, create a cube. Click the Channel Box/Layer Editor tab on the right of the UI to show the Channel Box. With the new cube selected, click the poly-Cube2 entry (your cube may have a slightly different number, like polyCube3 or so), which will show you attributes for the cube's subdivisions. Set Subdivisions Width to 2, Subdivisions Height to 2, and Subdivisions Depth to 6, as shown in Figure 6.28.
- 3. Turn the display toggle of the pontoonRefLayer back to R. Move, rotate and scale the cube to fit the gray base of the attaching arms in the top view panel. Move its vertices to better match the shape, as shown in Figure 6.29. Note that this base shape is purposely not centered on the pontoon. Name the cube pontoonArmBase. Also, select the body of the pontoon and name it **pontoon**.



Figure 6.27 **Enable Modeling Toolkit.**



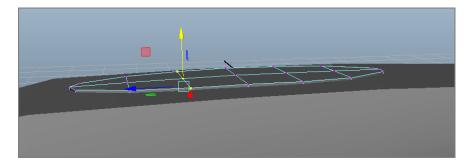
Create a new cube with these subdivision values.



Figure 6.29 Shape the cube to fit the base of the attaching arms.

4. Turn off X-Ray mode if it's on in the persp view panel (Shading \rightarrow X-Ray). Move the pontoonArmBase cube and scale it down in the Y-axis to have just the top sticking through the pontoon. Adjust the top vertices to curve the top to fit the top of the pontoon, as shown in Figure 6.30.

Figure 6.30 Scale and position the cube. Adjust the top vertices to match the pontoon's curvature.

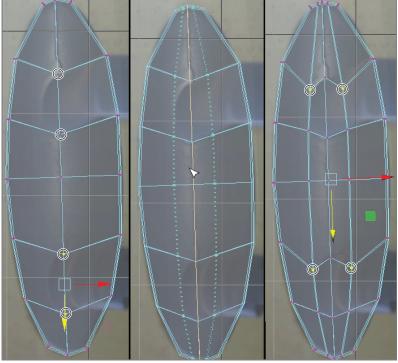


5. Click the Modeling Toolkit tab (on the right of the UI) to display it instead of the Channel Box or Attribute Editor. Click the Edge Selection icon at the top of the toolkit (). As you hover your cursor over edges of the pontoonArmBase cube, they turn red. Select the outside loop of edges all the way around the top, as shown in Figure 6.31 (top). Then click the Bevel button in Modeling Toolkit. Enter a Segments value of 2 and an Offset value of 0.60, as shown in Figure 6.31.

Transform Constraints ▼ Mesh Editing Tools Nevel 🎇 Target Weld

Figure 6.31 Select the top outside edges (top) and bevel them using Modeling Toolkit.

6. Click the Bevel button again to exit the tool. In the top view, adjust the shown vertices to align with the arms coming out of the base, as shown in Figure 6.32 (left). Then in the main menu bar, choose Mesh Tools → EDIT → Offset Edge Loop Tool and select the middle vertical loop of edges on top of the arm base to place two edge loops, as shown in Figure 6.32 (middle). Press W to exit the tool and enter the Move tool. Select the shown vertices and adjust them to fit the base of the arms, as shown in Figure 6.32 (right).



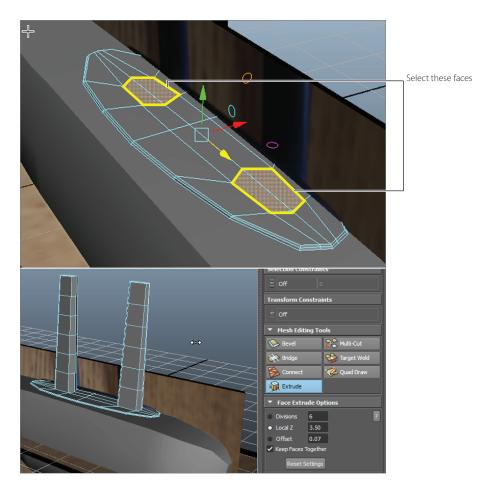
Adjust these vertices

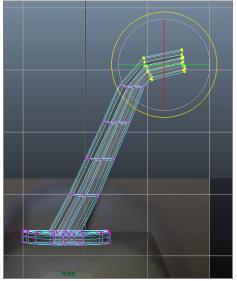
- 7. In Modeling Toolkit, click the Face Selection icon (🔠). Select the faces shown in Figure 6.33 (top) and then click the Extrude button in Modeling Toolkit. First set the Local Z value to 3.50 to extrude the faces up in space, next set Divisions to 6 for extra segments, and finally set Offset to 0.07 for a slight taper, as shown in Figure 6.33 (bottom). Click the Extrude button to exit the tool.
- 8. In the front view panel, select vertices and move them to create an angle to the right for the arm. At the top of the arm, select the top two rows of vertices and move and rotate them to create a bend, as shown in Figure 6.34.

Figure 6.32

Line up the vertices to where the arms come out of the base (left) and then use Offset Edge Loop to place edge loops as shown (middle). Finally, adjust the selected vertices to fit around the arm at the base (right).

Figure 6.33
Select these faces (top) and extrude them with Modeling Toolkit (bottom).





- 9. Click the Channel Box/Layer Editor tab to switch out of the Modeling Toolkit view, and in the Layer Editor, turn off the visibility of the pontoonRefLayer. Turn off X-Ray view in your views.
- 10. Select both the pontoon and the arm base and press Ctrl+G to group them. Name the group **leftPontoonGroup**. Press 3 to see the pontoon in Smooth Preview, as shown in Figure 6.35. Save your work. There's just one detail to add at the back of the pontoon.

Figure 6.34
Create an angle to the arm and a bend at the tip.

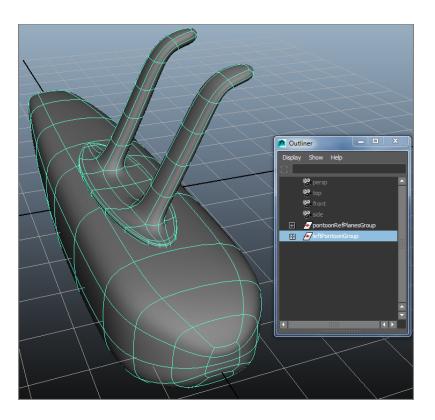
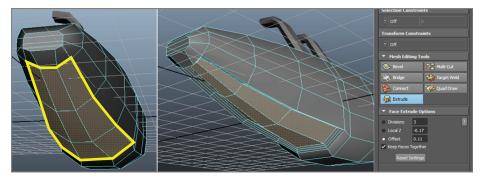


Figure 6.35

The left side pontoon is almost done and shown here in smooth preview.

11. With the pontoon selected, press 1 to exit smooth preview. Go back into Modeling Toolkit by clicking its tab. Enter Face selection () and select the nine faces in the back-bottom part of the pontoon, as shown in Figure 6.36 (left). Click the Extrude button in Modeling Toolkit and set Divisions to 3, Local Z to -0.17, and Offset to **0.11**, as shown in Figure 6.36 (right). Click the Extrude button to exit the tool. This creates a little indent at the back of the pontoon, and the Divisions value of 3 will keep the edges sharper when you eventually smooth the model.



Select these faces

Figure 6.36 Select these nine faces (left) and create an inset in the bottom back of the

pontoon (right).

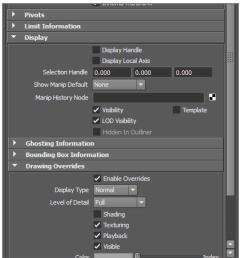
- 12. Click the Power Switch icon at the top right of the Modeling Toolkit interface (**1**) to deactivate Modeling Toolkit for the time being. The icon will go from blue to gray.
- 13. In the Outliner, select the pontoonRefPlanesGroup and delete it. In the Layer Editor, RMB+click the pontoonRefLayer and select Delete Layer. Lastly, open the Hypershade (Window → Rendering Editors → Hypershade) and select Edit → Delete Unused Nodes. This will get rid of the reference photos on the planes you just deleted. You are doing all this so when you assemble the final plane, you will not import unwanted reference planes, display layers, or shaders along with the pontoon. It's best to keep a clean workflow!

You're done with the pontoon! Go celebrate with some cookies. Save your work! You will undoubtedly adjust this model later when you attach it to the plane's body, as well as smooth the geometry. You can compare your work to the scene file pontoonModel_v04.ma in the Scenes folder of the Plane project.

Oh, What a Body! Modeling the Body of the Plane

With one of the pontoons finished, you can move on to the body of the plane in much the same way you modeled the pontoon. If you like a challenge and can use the practice, use the steps at the beginning of the chapter where you created three reference view planes for the pontoon to create three new reference view planes for the body of the plane in a new Maya scene. In the Sourceimages folder of the Plane project, you will find bodyBackReference.jpg, bodyBottomReference.jpg, and bodySideReference.jpg to use as the images. They are all sized 1600×1200 with an aspect ratio of 1:0.750. Just make sure to line up the planes and uniformly scale them as needed. Group them together and make sure the reference planes end up about 14 units in length to better fit the scale of the pontoon you already made.

Figure 6.37 Create a drawing override to make the cube display as a wireframe all the time.



You can skip creating the reference planes and get right to modeling by loading the scene file bodyModel_v01.ma in the Scenes folder of the Plane project. This file has the reference view planes already set up and ready with the proper grouping and scale. See how nice I am?

Blocking Out the Shape

In this section, you'll block out the shape of the body; follow these steps:

1. Create a cube at the origin. This time instead of using X-Ray shading to see through the model to the reference images, you'll force the body geometry to show as wireframe even in Shaded mode. With the cube selected, open the Attribute Editor, click the pCube1 tab, and then click the Display heading. Click the Drawing Overrides heading, check the box Enable Overrides, and then uncheck the Shading box, as shown in Figure 6.37.

Your cube should display as a wireframe, even though your reference planes display Textured view.

- 2. Using the top and side views, scale and place the cube over the body of the plane. You want a little overlap between the body and the gray engine at the front of the plane, as shown in Figure 6.38. In the Channel Box, click the polyCube1 entry to reveal the subdivision values.
- 3. In the side view panel, insert eight vertical edge loops (Mesh Tools → EDIT → Insert Edge Loop Tool), as shown in Figure 6.39 (top). Still in the side view, adjust the vertices to shape the cube to the plane, as shown in Figure 6.39 (bottom).

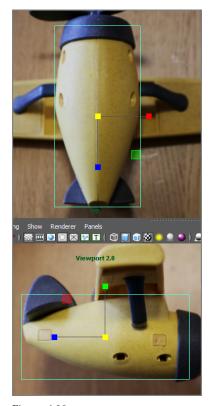
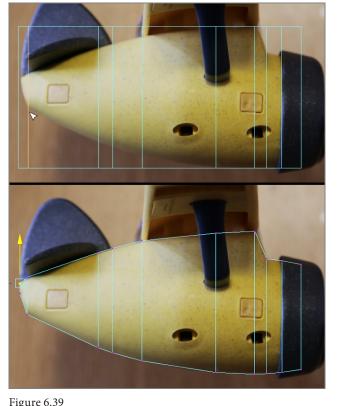
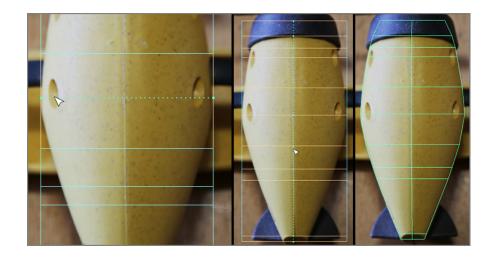


Figure 6.38 Create a cube and place it properly using the top and side views.

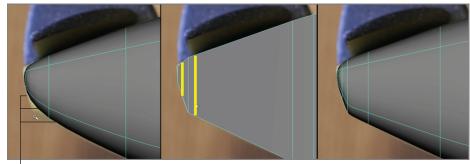


Insert these edge loops (top) and then adjust the vertices to shape the cube to the plane (bottom).

Place a horizontal edge loop (left) and then a vertical edge loop up the center (middle). Then move vertices to shape the body in the top view (right).



- 4. In the top view, insert a horizontal edge loop at the widest part of the plane, as shown in Figure 6.40 (left). Then place a vertical edge loop straight up the middle of the plane, as shown in Figure 6.40 (middle). Finally, adjust the vertices to fit the shape of the plane, as shown in Figure 6.40 (right). Name the object **body**.
- 5. Go into the persp view panel, and with the body selected, press 3 for smooth preview. In the Attribute Editor for the body, under Display → Drawing Overrides, turn off Enable Overrides to see the mesh in Shaded mode. It's looking pretty good.
- 6. In the side view panel, notice the very back end of the body is too rounded when you compare it to the side of the plane, as in Figure 6.41 (left). Press 1 to exit smooth preview, and it lines back up again. The smoothing rounds that corner too much. To avoid that, you need to add two more edge loops, as shown in Figure 6.41 (middle), to preserve detail when smoothed. Press 3 again, and the back end lines up nicer in smooth preview, as shown in Figure 6.41 (right).
- 7. Turn off the image reference planes' layer (bodyRefLayer) in the Layer Editor.



Too rounded here

Insert these edge loops

Figure 6.41
The end of the body gets too smooth (left), so add these two edge loops (middle), and when you smooth preview again, the end fits better (right).

8. Press 1 to exit smooth preview. Turn Modeling Toolkit back on by clicking its tab and then clicking the Power button () to turn it blue.

If Modeling Toolkit tab is not present, you need to click the Show Modeling Toolkit icon (M) in the upper right of the UI (as previously shown in Figure 6.27).

9. Enable Face selection by clicking its icon () and select the six faces at the back of the body, as shown in Figure 6.42 (left). Then click the Extrude button and set Divisions to 3, Local Z to -0.10, and Offset to 0.20. This creates a little inset at the back of the plane similar to the toy's back end. Click the Extrude button again to exit the tool.

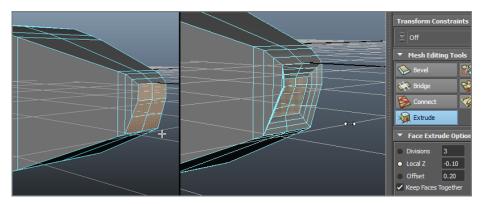


Figure 6.42 Select these faces (left) and then extrude them to

create an inset.

10. With the body selected, press 3 for smooth preview, and you'll notice a little bit of pinching around the inset you just created as in Figure 6.43 (left). Stay in smooth preview. Enter vertex selection by clicking the 🔀 icon in Modeling Toolkit and move a pair of vertices, as shown in Figure 6.43 (right), to alleviate the pinching. Repeat for the other side.

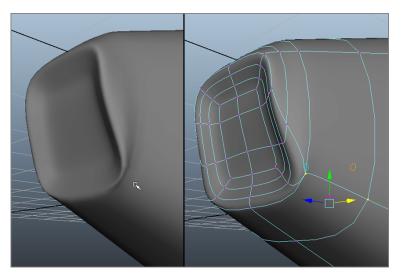


Figure 6.43

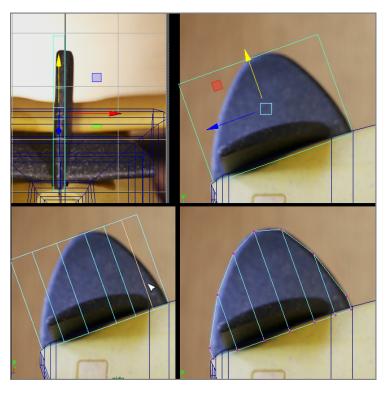
There is a pinch in the back (left), so adjust a few vertices to fix it (right).

Now you're done with the body of the plane. You can check your work against the scene file bodyModel_v02.ma in the Scenes folder of the Plane project.

The Rear Stabilizers

You'll create these back stabilizer wings as a new model directly in the body scene. Continue with your file or use bodyModel_v02.ma to start modeling the rear stabilizer wings.

- 1. Turn visibility on for bodyRefLayer in the Layer Editor (click the toggle to display a V). Select the body geometry and turn on Enable Overrides again to make it display as wireframe (found in Attribute Editor → Object Display → Drawing Overrides) as you did in step 1 of the previous section, "Blocking Out the Shape."
- 2. Create a poly cube in the side view panel; using the side and front view panels, move, rotate, and scale the cube to roughly fit the vertical stabilizer wing, as shown in Figure 6.44 (top left and top right).



3. Using the Insert Edge Loop tool, insert five new edge loops vertically, as shown in Figure 6.44 (bottom left), and use the vertices to adjust the shape of the wing, as shown in Figure 6.44 (bottom right).

Figure 6.44

Create and place a cube for the vertical wing (top). Insert edge loops (bottom left) and adjust the shape (bottom right).

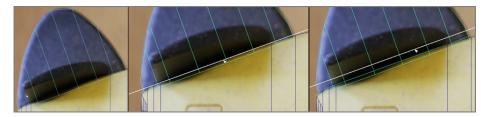


Figure 6.45

Insert a vertical edge loop (left) and then use Cut Faces to insert a pair of horizontal edges that loop around (middle and right).

5. Now you need to insert some edge loops to let you extrude the horizontal side wings that come out of the vertical wing to make an upside-down *T* for the stabilizers. If you use the Insert Edge Loop tool, the edge loops will be curved like the arch of the shape so far. Instead, select the vertical wing object, and choose Mesh Tools → EDIT → Cut Faces Tool. This tool allows you to generate a loop of edges that will be oriented by how you click in the view panel. Your cursor turns to a triangle.

In the side view panel, click the vertical wing object where the yellow plane body meets the gray vertical wing and drag until the white line that appears lines up, as shown in Figure 6.45 (middle). As soon as you release the mouse, Maya inserts edges that loop around at the base of the vertical wing.

- 6. Use the tool again to place a parallel cut a little further up the vertical wing, as shown in Figure 6.45 (right).
- 7. In Modeling Toolkit, enable edge selection (). In the persp view panel, if you double-click one of the upper edges on the vertical wing's top arch, it will select an entire edge loop on the top arch of the wing shape. Do this twice to select the edge loops on both sides of the arch of the wing, as shown in Figure 6.46 (left).

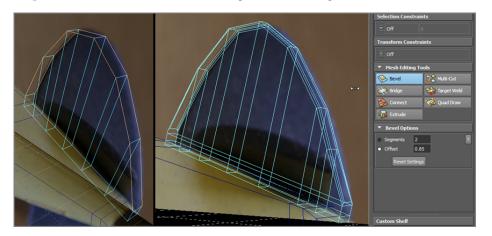
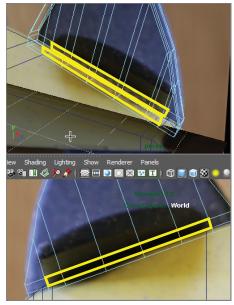


Figure 6.46
Select these edges (left) and bevel them (right).

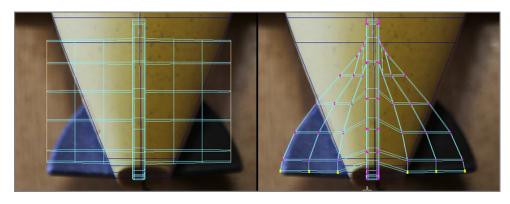


Select five faces on both sides

Figure 6.47 Select these five faces on both sides of the vertical wing.

- 8. Click the Bevel button in Modeling Toolkit and set Segments to 2 and Offset to 0.85, as shown in Figure 6.46 (right). Click Bevel again to exit the tool.
- 9. In the persp view panel, select the five faces shown at the base of the vertical wing where the side wings are. Make sure to select five on each side of the vertical wing, as shown in Figure 6.47.
- 10. In Modeling Toolkit, click Extrude and set Divisions to 3, Local Z to 1.5, and Offset to 0.03 for a slight taper at the end, as shown in Figure 6.48 (left). Click Extrude to turn it off. Then adjust the vertices in the top view to adjust vertices to loosely match the shape of these side wings for the stabilizers, as shown in Figure 6.48 (right). As you can see, the image used for reference is not centered well, but keep your geometry even, even though it won't line up well on the right side of the plane in the image.
- 11. Turn off the visibility of the bodyRefLayer in the Layer Editor. For the stabilizer wings, uncheck Enable Overrides in the
- Attribute Editor to show the wings in Shaded mode. Also, in the persp view's panel menu bar, select Shading - Wireframe On Shaded to display wireframe on top of the shaded models.
- 12. Select the outer loop of edges around the side wings, as shown in Figure 6.49 (left), and click Bevel in Modeling Toolkit with a Segments value of 2 and an Offset value of 1.5, as shown in Figure 6.49 (right).
- 13. At the front of the vertical wing, select the vertices shown in Figure 6.50 and scale them closer together in the *X*-axis to taper the front blade of that wing a bit. Select the tail wing object and call it tail.

Figure 6.48 Extrude the side wings (left) and shape them to create a nice shape, even though it won't line up on one side (right).



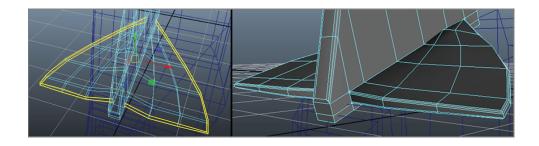


Figure 6.49 Select the edges around the side wings (left) and bevel them (right).

14. Select the plane's body geometry and uncheck Enable Overrides (Attribute Editor → Object Display → Drawing Overrides) so you can see the entire plane in Shaded mode. Press 3 to see it in smooth preview. Press 1 to get out of smooth preview for the meshes.

Save your work! You can check your work against the scene file bodyModel_v03.ma in the Scenes folder of the Plane project.

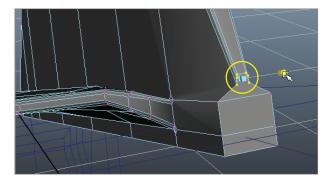
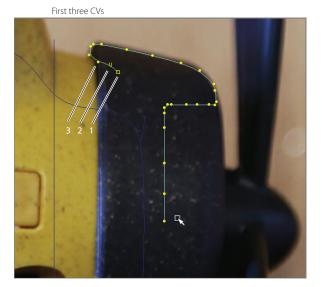


Figure 6.50 Select these vertices and squeeze them together in the X-axis.

You Spin Me Right Round—The Engine and Propeller

Now for the propeller! In the same scene as the body, you'll create the engine and propellers at the front of the plane. Begin with your current scene, or use bodyModel_v03.ma. To begin with the engine, follow along here:

- 1. In the Layer Editor, make sure the bodyRefLayer is visible and the R toggle is off so you can see and select the plane images. You need only the side reference image now, so select the other two planes and press Ctrl+H to hide them. In the Outliner, their entries (bodyBottomRefPlane and bodyBackRefPlane) show up as gray. To unhide them, you would select them in the Outliner and press Shift+H. Leave them hidden for now.
- 2. Select the body and press 3 for smooth preview.
- 3. In the side view, create a CV curve to match the top half profile of the engine, as shown in Figure 6.51. You'll use this to revolve a surface.
- 4. In the persp view, select the profile curve and choose Display → NURBS → CVs to show the CVs of the curve. Click the Snap To Points icon in the status bar (\sum_1).
- 5. Press and hold D to engage the Pivot Move tool and move the pivot of the curve to snap to the bottom CV in the middle of the engine, as shown in Figure 6.52. This will ensure a good revolve. Turn off the point snap and choose Display → NURBS → CVs to turn off the CVs on the curve.



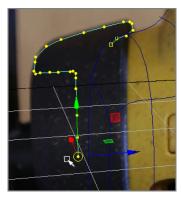


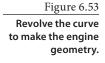
Figure 6.51

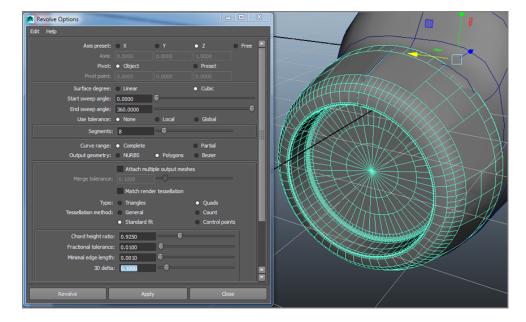
Draw a profile curve for the shape of the engine.

Figure 6.52

Snap the pivot to the bottom CV.

6. Switch to the Surfaces menu set. Then, with the curve selected, choose Surfaces → Revolve □. Set Axis preset to Z, Segments to 8, Output Geometry to Polygons, and Type to Quads; leave the Tessellation method on Standard Fit. Set the Chord Height ratio to 0.925 and the 3D delta to 0.10. Click Revolve to make the engine fit around the front of the plane, as shown in Figure 6.53.





7. The engine may not be perfectly centered on the plane body, so move it to center it on the body. Then choose Edit → Delete By Type → History to erase the history on that shape. Select the profile curve and delete it. Select the engine geometry you just created and name it **engine**.

Engine Detail

Now for some detail inside the engine.

- 1. Choose Create → Polygon Primitives → Cylinder □ and set Axis Divisions to 16, Height Divisions to 11, and Cap Divisions to 1. Set Axis to Y and leave the rest at the defaults. Click Create and place the cylinder in front of the engine. Scale it to **0.125** in *X* and Z, and set the Y-axis scale to **0.4**, as shown in Figure 6.54.
- 2. In Modeling Toolkit, enter face selection; in the side view, select every other ring of faces, as shown in Figure 6.55 (left). Click the Extrude button with an Offset value of **0.03** and a Divisions value of **1**, as shown in Figure 6.55 (right). Turn off Extrude.
- 3. Select the cylinder in Object mode, move it back into the engine, and place it as shown in Figure 6.56.

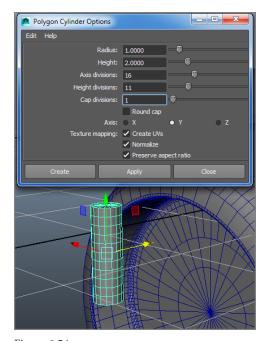


Figure 6.54 Create this cylinder and place it in front of the engine.

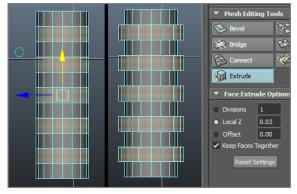


Figure 6.55 Select every other ring of faces (left) and extrude them (right).

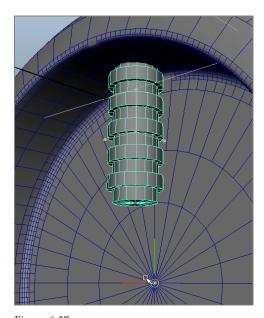


Figure 6.56

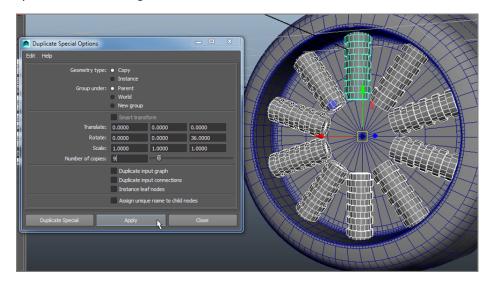
Place the cylinder into the engine.

Figure 6.57
Snap the pivot to the engine's center.

- 4. Turn on Snap To Points () and press and hold D to move the pivot. Snap the pivot point to the center of the engine, as shown in Figure 6.57.
- 5. Choose Edit → Duplicate Special □. Set Rotate to **36** in the *Z*-axis and Number Of Copies to **9** and click Apply, as shown in Figure 6.58. This creates a ring of these cylinders inside the engine.

Figure 6.58

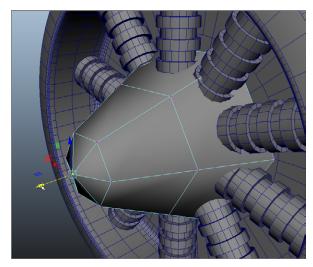
Create a ring of cylinders inside the engine.



The Propeller Blades

Follow these steps to create the propeller blades:

- 1. Create a polygon cone. In the Channel Box, click polyCone1 and set Subdivisions Axis to 6 and Subdivisions Height to 3. Scale the cone to 0.72 in all three axes and rotate and place it against the inside of the engine. Select the sharp tip's vertex and push it back to make the cone blunt, as shown in Figure 6.59.
- 2. Switch back to the Polygons menu set and insert an edge loop toward the middle of the cone, as shown in Figure 6.60. Select every other of the new faces on the cone and click the Extrude button in Modeling Toolkit. Set Divisions to 1, Local Z to 3.2, and Offset to **0** to create the blades of the propeller, as shown in Figure 6.61. Click Extrude to turn off the tool.
- 3. The blades sure do look weird; they are angled away from the engine. With the three faces at the ends of the blades still selected, move them back in the Y-axis to make them flush with the engine. The blades will go through the engine geometry a little bit, and that's okay for now.
- 4. On each of the three blades, insert an edge loop about where the blade intersects with the engine geometry, as shown in Figure 6.62 (left), and then scale the new edge loops to be much larger, as in Figure 6.62 (right).
- 5. If you press 3 for smooth preview, the propellers look pretty good. Select them and move them out so the blades don't penetrate the engine geometry.
- 6. Stay in smooth preview for the body (but not the engine; the engine will not need smoothing later) and work around the body of the plane. Move vertices on the body where it meets the engine to make sure that the body is not penetrating the engine.





Place a cone in the engine and shorten the tip to make it blunt.

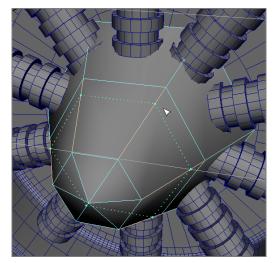


Figure 6.60

Insert an edge loop.

Figure 6.61

Select every other face (left) and extrude them to create the propeller blades (right).

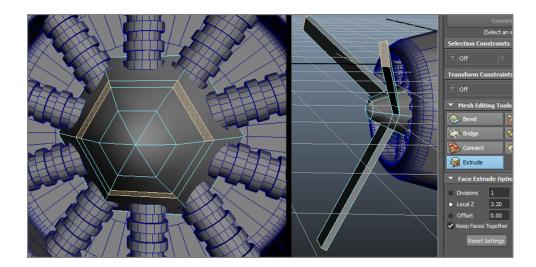
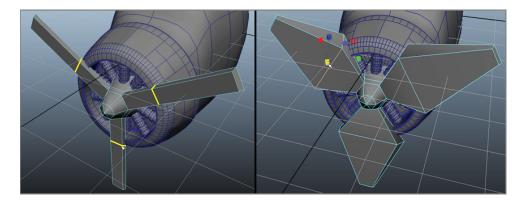


Figure 6.62

Insert an edge loop on each of the three blades (left) and then scale the new edge loops to be much bigger (right).



Now you're done with the body and need to move to the wings. Save your work and compare it to the scene file bodyModel_v04.ma in the Scenes folder of the Plane project. Note that this file has the plane body and propeller and stabilizer wings already set in smooth preview.

The Plane's Wings

How can you fly without wings? Here you'll create the wings in the existing plane body scene file and get them to attach to the body.

Modeling the Wing

Continue with your own model or use bodyModel_v04.ma to continue.

1. Turn the visibility on for the bodyRefLayer. Then, in the Outliner, select the two hidden gray reference planes grouped under bodyRefPlanesGroup (click the plus sign to reveal them in the Outliner). Press Shift+H to unhide them in your view panels.

Figure 6.63

Create a cube and place it as shown to

start the left side

of the wing (top). Insert the edge

loops as shown

- 2. Select all the parts of the plane you have made so far and press Ctrl+H to hide them.
- 3. In the persp view, press 6 for Texture mode to see the reference of the toy plane again. Create a cube and, in the Attribute Editor, check Enable Overrides under Object Display → Drawing Overrides. Turn off Shading as you've done before to make the cube display as wireframe. Using the top views, size and orient the cube to fit over the wing, as shown in Figure 6.63 (top). Since there is some perspective shift in the photos, the placements will be rough and will not line up in the side view at all, and that's okay. Insert edge loops, as shown in Figure 6.63 (bottom), to create more segments.
- 4. Now you'll need to shape the wing. This time, let's use a feature called Symmetry. Turn off Modeling Toolkit by clicking the Power icon () if it's still on and glowing blue. In the top view, RMB+click the body and enter vertex mode. Then choose Modify → Transformation Tools → Move Tool \square (or double-click the Move tool icon $[\![\, \downarrow \,]\!]$ in the Tool Box).
- 5. In the option box, toward the bottom, turn on Symmetry. Now, any time you select vertices on one side of the wing, the opposite side's vertices will be selected as well. This way you can shape one side of the wing, and the other side will shape itself to match.
- 6. Use the top view to shape the left side of the wing, and as you do so, the right side will shape itself. Don't worry if the shape of the wing does not match the image; remember that the image is not perfectly symmetrical. This is okay. Figure 6.64 shows the wing shaped using the Reflection feature.
- 7. In the front view and with Symmetry still on, shape the wing on the left to the image, as shown in Figure 6.65 (top).

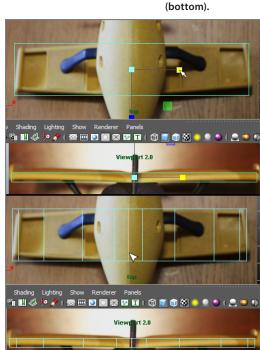
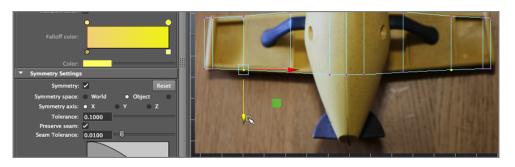
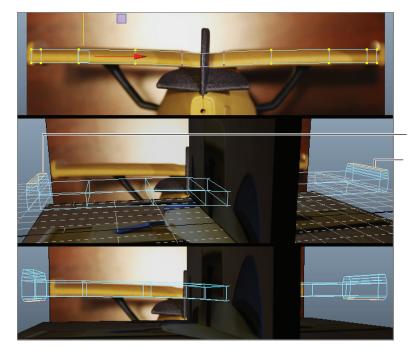


Figure 6.64

Using Symmetry lets you shape one side of the wing and reflect that shape to the other side automatically.

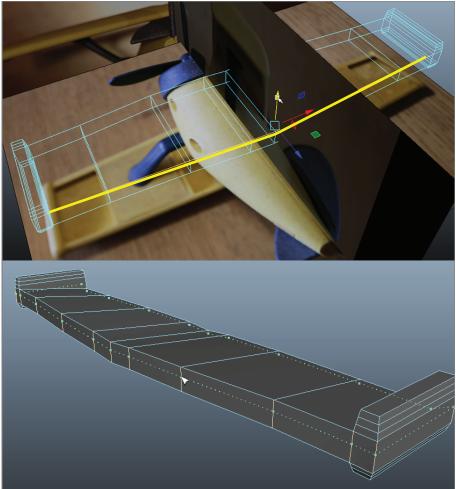


Shape the vertices on one side, and Symmetry will shape the other side (top). Extrude the top face (middle). Then select and extrude the bottom end face (bottom).



Select this face and extrude Symmetry will select and extrude this face automatically

- 8. RMB+click the body and choose Faces for face selection mode. At the end of the wing on the left, select the top face at the very end. Symmetry will automatically select the other side's top end face as well.
- 9. Then turn Modeling Toolkit back on and click the Extrude button. Set Divisions to 3, Local Z to 0.35, and Offset to 0.05. Both ends will extrude up, as shown in Figure 6.65 (middle).
- 10. Turn off Modeling Toolkit to allow Symmetry to work again and then select the bottom face at the end of the wing on the left side. Again, in Modeling Toolkit, click the Extrude button. Set Divisions to 2, Local Z to 0.15, and Offset to 0.05. This will extrude the ends of the wings, as shown in Figure 6.65 (bottom).
- 11. Using Modeling Toolkit will have turned off Symmetry, which is okay. Turn off display of the reference image layer (bodyRefLayer). From here on, you'll shape the plane without the reference images. Select the top back edge that runs across the wing, as shown in Figure 6.66 (top), and move the edges down to taper the back of the wing down.
- 12. Select the wing and in the Attribute Editor uncheck Enable Overrides to show the wing in Shaded mode again. Press 5 for Shaded mode in the persp view. Insert an edge loop two-thirds of the way up the side of the wing, as shown in Figure 6.66 (bottom). This will help retain the shape of the wing when you smooth the mesh later.

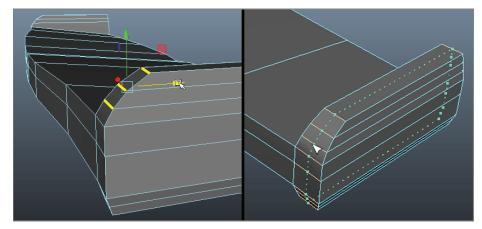


Select the edges on the top back of the wing

- 13. Open the option box for the Move tool (Modify → Transformation Tools → Move Tool □). Turn on Symmetry.
- 14. RMB+click the wing and enter edge selection. Shape the tip of the wing as shown in Figure 6.67. Symmetry will take care of the opposite side. Then add an edge loop as shown. You will have to add an edge loop to the wing tip on the other side; Symmetry will not work here.
- 15. Make any personal adjustments you'd like and press 3 to smooth preview the wing. Once you're happy with the shape, press 1 to exit smooth preview and choose Mesh → SHAPE → Smooth □. Make sure the settings are at default in the option box by selecting Edit → Reset Settings and then click Smooth.

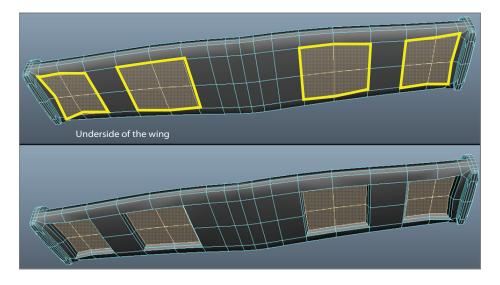
Figure 6.66

Top: Select the topside back edges of the wing and move them down to taper the wing. Bottom: Insert an edge loop two-thirds up the side of the wing. Shape the tip of the wing to curve it back a bit (left) and then add an edge loop at the wing tip for both sides of the wing (right).



Move these edges to shape the tip

16. Orbit your camera to look at the underside of the wing. On the underside of the wing, select the faces, as shown in Figure 6.68. In Modeling Toolkit, click the Extrude button. Set Divisions to 2, Local Z to -0.16, and Offset to 0.10, as in Figure 6.68 (bottom). Turn off Extrude.



under the wing (top) and extrude to create indents in the wing's underside (bottom).

Figure 6.68
Select these faces

17. Now select the outer edges of those four new indents in the underside of the wing, as shown in Figure 6.69 (top). In Modeling Toolkit, bevel them with a Segments value of **2** and an Offset value of **0.60**, as shown in Figure 6.69 (bottom). Turn off Bevel. Save!

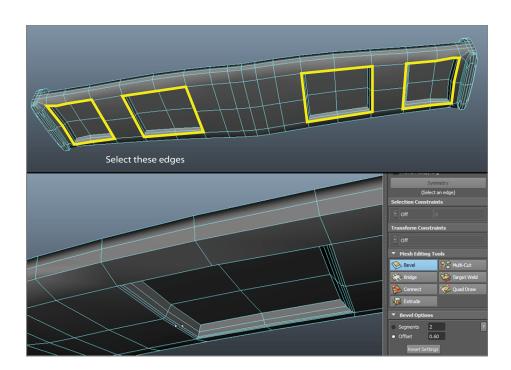


Figure 6.69

Select these edges (top) and bevel them (bottom).

Organizing the Scene and Smoothing the Geometry

Now let's take a moment to clean up and organize the scene and smooth the rest of the geometry:

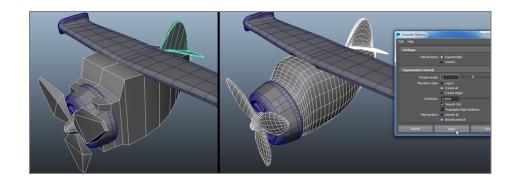
- 1. Open the Outliner and select all the hidden objects that appear in gray text. Press Shift+H to unhide them. You should now see the plane, back stabilizer wings, and propellers in smooth preview.
- 2. In the Outliner, select all the pCylinder objects you created inside the engine and press Ctrl+G to group them. Call the group insideEngine.
- 3. Select the propeller and name it **props**. Select the wing and call it **wings**. Figure 6.70 shows the Outliner with the proper names.
- 4. Select all the parts of the plane and press 1 to exit smooth preview for all the objects.
- 5. Select the body, propellers, and tail wings, as shown in Figure 6.71 (left), and choose Mesh → SHAPE → Smooth □. Check to make sure you are set to Exponentially for Add Divisions and set Division Levels to 2. Click Apply, as shown in Figure 6.71 (right).

Figure 6.70

The Outliner shows the organized



The plane before smoothing (left) and after smoothing (right)



- 6. Select all the geometry and choose Edit → Delete By Type → History to get rid of all the construction history.
- 7. Since you don't need the reference images anymore, delete the reference plane's top group (bodyRefPlanesGroup). RMB+click the bodyRefLayer in the Layer Editor and select Delete Layer. And in the Hypershade (Window → Rendering Editors → Hypershade), click Edit → Delete Unused Nodes. Save!

Adding the Wing Supports

Now you just need to connect the wings to the plane using the following steps:

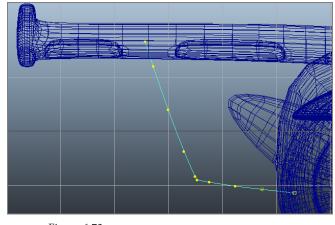


Figure 6.72
Create this curve for the wing support.

- The wings sit a little low and penetrate the top of the plane's body a little bit. Select the wing object and move it up to sit on top of the body. Rotate it a bit in the X-axis to align it with the angle of the top of the plane's body better.
- 2. Let's smooth the wing a little more. With the wing selected, choose Mesh → SHAPE → Smooth, set the Division level back to 1, and click Smooth. Select the smoothed wing and delete the history (Edit → Delete By Type → History).
- 3. Now let's make the support connecting the wing to the sides of the plane. Maximize the front view (press the spacebar). Choose Create → CV Curve Tool and create a curve starting at the body of the plane up to the underside of the left side wing, as shown in Figure 6.72.

- 4. In the persp view's menu, select Show and uncheck Polygons to turn off polygon view in the view panel. This shows you the curve you just made by itself. Select the curve and select Display → NURBS → CVs to show the CV points on the curve.
- 5. Next, create a circle (Create → NURBS Primitives → Circle). Scale it to a small oval and place it at the start of the CV curve line you just made, which is just inside the body of the plane. You can use Snap To Points to place the oval exactly at the start of the curve, as shown in Figure 6.73.
- In the persp view panel's menu, click Show and check Polygons back on. Stay in wireframe view (press 4).
- Select the oval and the curve and switch to the Surfaces menu set.
 Select Surfaces → Extrude □. Set Style to Tube, Result Position to At Path, Output Geometry to Polygons, and Type to Quads; click Extrude, as

Figure 6.73

Create an oval shape and place it at the start of the curve.

shown in Figure 6.74. Press 5 to see it in Shaded mode and assess how you like it. You can adjust the shape or size of the oval as well as the shape of the CV curve line to customize the resulting support shape that connects the wing to the body.

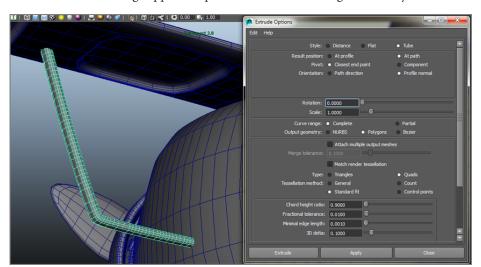
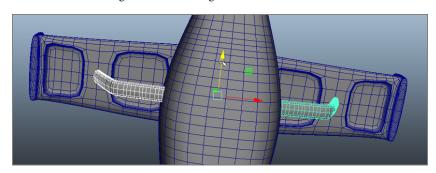


Figure 6.74

Extrude to create the support for the wing.

- 8. If you are happy with the shape of the support, select the original oval and the CV curve and delete them. Select the support arm geometry and name it supportArm in the Outliner.
- 9. Now you need another for the other side of the plane. With supportArm selected, press Ctrl+D to duplicate it in place. In the Channel Box, enter an X-Scale value of -1.0 to create a mirror of the original arm. Since the pivot point of the original support Arm is in the center, scaling it in *X*-axis to -1 should place it perfectly on the other side of the plane.
- 10. Move the arms a little toward the back of the plane so the part that meets the wing is centered on the wing, as shown in Figure 6.75.



- 11. Select both support arms and choose Modify → Freeze Transformations. Name them wingSupport1 and wingSupport2.
- 12. Select the wing supports and the wing itself, group them together, and call the group wingGroup. Move wingGroup a little closer to the front of the plane to line up the wing, as shown in Figure 6.76.
- 13. Select the insideEngine and wingGroup groups and all the other parts of the plane and group them together. Call the group planeGroup.

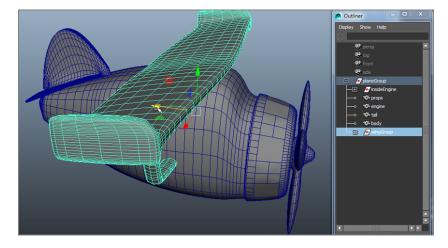


Figure 6.75 The support arms in place!

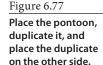
Figure 6.76 Place the wing assembly.

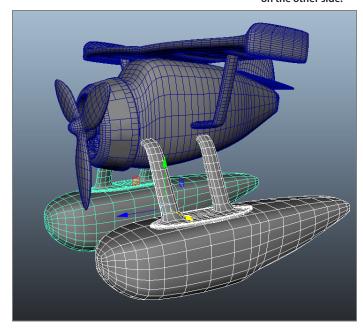
Save your work and compare it to the scene file bodyModel_v05.ma in the Scenes folder of the Plane project.

Assembling the Plane

You have all the parts finished, so let's assemble the plane into a complete scene and make any finishing touches. In your current body model or the bodyModel_v05.ma scene, you just need to import the pontoons and you're done.

- 1. Choose File → Import. In the file dialog, on the right side, make sure Use Namespaces is checked. Navigate to the latest pontoon model you created or to the pontoonModel_ v04.ma file in the Scenes folder of the Plane project on the website.
- 2. A single pontoon will appear, unsmoothed, in the middle of the scene. In the Outliner, its name will be something like pontoonModel_v04:leftPontoonGroup. The name before the colon is the filename from which you imported the model. The part of the name on the right is the node name.
- 3. Select the two pieces of geometry that make up the pontoon (not the top group) and choose Mesh → SHAPE → Smooth □. Set Division Levels to 1 and click Smooth.
- 4. Select the top node of the newly smoothed pontoon assembly (pontoonModel_ v04:leftPontoonGroup) and rename it to leftPontoonGroup in the Outliner. Then move it into place (as shown in Figure 6.77) and orient it properly by rotating it 180 degrees in the *Y*-axis.
- 5. Duplicate the pontoon group (leftPontoonGroup) by pressing Ctrl+D and set the copy's X-axis scale to -1.0. Call the new group rightPontoonGroup and move it to the other side of the plane body. Figure 6.77 shows the proper placement for both pontoons.
- 6. Select the geometry for both pontoons (not the top nodes) and choose Edit → Delete By Type → History.
- 7. With the geometry still selected, choose Modify → Freeze Transformations.
- 8. Using the Outliner, place both pontoon groups into the planeGroup by MMB+dragging them onto the plane-Group node.





- 9. Now you'll adjust how the pontoons look on the plane. In the Outliner, select the two top nodes of the pontoons. In the Animation menu set, select Create Deformers → Lattice □. Set Divisions to 4, 3, 2 and click Create. This places a lattice around the pontoons.
- 10. You want to spread the pontoons to make a wider base for the place. RMB+click the lattice and choose Lattice Point. Select the eight lattice points shown in Figure 6.78 (left) and move them away from the plane and up closer to the wing, as shown in Figure 6.78 (right).

Select these eight lattice points

- 11. Repeat for the other pontoon's lattice points to make the stance of the pontoons wider and closer to the wing. If your pontoon geometry is different from the one shown or the ones in the sample scene file pontoonModel_v04.ma, you may need to adjust your lattice differently. You may also choose to skip this step because it's a minor tweak. The proportions look nicer now!
- 12. Select the pontoon group nodes (Figure 6.79) and delete their history to get rid of the lattice (Edit → Delete By Type → History).

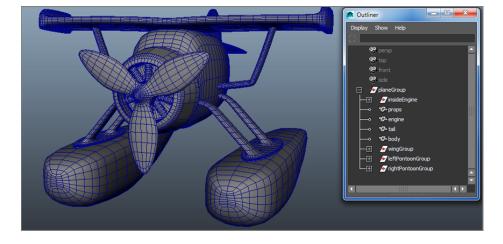


Figure 6.78

Select these eight lattice points and move the pontoon away from the plane body and up closer to the wing (left).

Figure 6.79

The pontoons are adjusted, and the plane is ready to fly! Figure 6.79 shows the completed plane and its hierarchy. You can check your work against the scene file bodyModel_v06.ma in the Scenes folder of the Plane project. In the following chapters, you will add textures and lighting to render the plane as a plastic bathtub toy.

Summary

In this chapter, you flexed your knowledge from the previous chapters and concentrated on creating a model of a child's bathtub toy airplane. You used many of the tools discussed in the previous chapters, from extruding to adding edge loops to using bevels.

Creating a model can be a lot of hard and sometimes tedious work, but when you start seeing it take shape, the excitement begins to build. From the basic shaping of the plane's parts to the detail of assembling the plane, you worked hard in this chapter to create the toy plane.

You can take the procedures you used in this chapter to build your own toy design. The important lesson to take away from this chapter is how in depth you can get with a model and how a lengthy modeling process takes shape. Along the way, don't forget to name your pieces and group everything in a sensible fashion.



Autodesk® Maya® Shading and Texturing

Shading *is the term* for applying colors and textures to create *materials*, also known in the Autodesk® Maya® software as *shaders*. A shader defines an object's look—its color, tactile texture, transparency, luminescence, glow, and so forth.

Learning Outcomes: In this chapter, you will be able to

- Differentiate between different shader types
- Create and edit shader networks in the Hypershade window
- Apply shaders and textures to a model
- Set up UVs on a model for the best texture placement
- Understand the steps to set up texture images to fit a model's UV layout
- Tweak UVs to align texture details
- Use Maya's toon shading to create a cartoon look

Maya Shading

When you create any objects, Maya assigns a default shader to them called Lambert1, which has a neutral gray color. The shader allows your objects to render and display properly. If no shader is attached to a surface, an object can't be seen when rendered. In the Maya view panels, it will appear wireframe all the time, or as flat green, even in Shaded view.

Shading is the proper term for applying a renderable color, surface bumps, transparency, reflection, shine, or similar attributes to an object in Maya. It's closely related to, but distinct from, texturing, which is what you do when you apply a map or other node to an attribute of a shader to create some sort of surface detail. For example, adding a scanned photo of a brick wall to the Color attribute of a shader is considered applying texture. Adding another scanned photo of the bumps and contours of the same brick wall to the Bump Mapping attribute is also considered applying a texture. Nevertheless, because textures are often applied to shaders, the entire process of shading is sometimes informally referred to as texturing. Applying textures to shaders is also called texture mapping or simply mapping.

Shaders are based on nodes. Each node holds the attributes that define the shader. You create shader networks of interconnected shading nodes, akin to the hierarchies and groups of models.

As you learn about shading in this chapter, you'll deal at length with the Hypershade window. See Chapter 2, "Jumping into Basic Animation Headfirst," and Chapter 3, "The Autodesk® Maya® 2015 Interface," for the layout of this window and for a hands-on introduction. You can access the Hypershade window by choosing Window → Rendering Editors → Hypershade. Shading in Maya is almost always done hand in hand with lighting. At the least, textures are tweaked and edited in the lighting stage of production.

Shader Types

Open the Hypershade window by choosing Window → Rendering Editors → Hypershade. In the left column of the Hypershade window, you'll see a listing of Maya shading nodes (Figure 7.1). The first section displays surface nodes, a.k.a. material nodes or shader types. Of these shader types, five are common to other animation packages. You'll use two of them later in this chapter.

The way light bounces off an object defines how you see that object. The surface of the object may have pigments that affect the wavelength of light that reflects off it, giving the surface color. Other features of that object's surface also dictate how light is reflected.

Most light, after it hits a surface, diffuses across an area of that surface showing off the object's color. It may also reflect a hot spot called a specular highlight (a basic reflection of a light source). The shaders in Maya differ in how they deal with specular and diffuse parameters according to the specific math that drives them. As you learn about the shader types, think of the things around you and what shader type would best fit them. Some Maya shaders, such as the Hair Tube shader and the Use Background shader, are

Figure 7.1 The Maya shading nodes



specific to creating special effects. It's important to learn the fundamentals first, so I'll cover the shading types you'll be using right off the bat.

The Lambert Shader Type

The default shader type is *Lambert*, an evenly diffused shading type found in dull or matte surfaces. A sheet of regular paper, for example, is a Lambert surface.

A Lambert surface diffuses and scatters light evenly across its surface in all directions, as you can see in Figure 7.2.

The Phong Shader Type

Phong shading brings to a surface's rendering the notions of specular highlight and reflectivity. A Phong surface reflects light with a sharp hot spot, creating a specular highlight that drops off sharply, as shown in Figure 7.3. You'll find that glossy objects such as plastics, glass, and most metals take well to Phong shading.

The Blinn Shader Type

The *Blinn* shading method brings to the surface a highly accurate specular lighting model that offers superior control over the specular's appearance (see Figure 7.4). A Blinn surface reflects light with a hot spot, creating a specular that diffuses somewhat more gradually than a Phong. The result is a shader that is good for use on shiny surfaces and metallic surfaces.

The Phong E Shader Type

The Phong E shader type expands the Phong shading model to include more control over the specular highlight. A Phong E surface reflects light much like a regular Phong does, but it has more detailed control over the specular settings to adjust the glossiness of the surface (see Figure 7.5). This creates a surface with a specular that drops off more gradually and yet remains sharper than a Blinn. Phong E also has greater color control over the specular than do Phong and Blinn, giving you more options for metallic reflections.



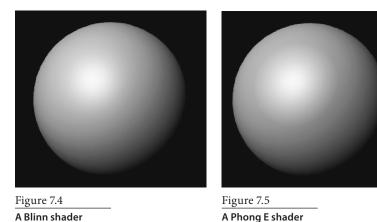
Figure 7.2

A Lambert shader



Figure 7.3

A Phong shader



The Anisotropic Shader Type

The Anisotropic shader is good to use on surfaces that are deformed, such as a foil wrapper or warped plastic (see Figure 7.6).

Anisotropic refers to something whose properties differ according to direction. An anisotropic surface reflects light unevenly and creates an irregular-shaped specular highlight that is good for representing surfaces with directional grooves, like CDs. This creates a specular highlight that is uneven across the surface, changing according to the direction you specify on the surface, as opposed to a Blinn or Phong where the specular is circular.

The Layered Shader Type

A Layered shader allows the stacking of shaders to create complex shading effects, which is useful for creating objects composed of multiple materials (see Figure 7.7). By using the Layered shader to texture different materials on different parts of the object, you can avoid using excess geometry.

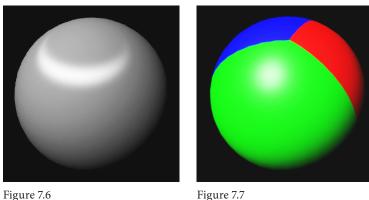


Figure 7.6 An Anisotropic shader

A Layered shader

You control Layered shaders by using transparency maps to define which areas show which layers of the shader. You drag material nodes into the top area of the Attribute Editor and stack them from left to right, with the leftmost shader being the topmost layer assigned to the surface, like a candy shell around a chocolate.

Layered shaders are valuable resources to control compound and complex shaders. They're also perfect for putting labels on objects or adding dirt to aged surfaces.

The Ramp Shader Type

A *ramp texture* is a gradient that can be attached to almost any attribute of a shader as a *texture node*. Ramps can create smooth transitions between colors and can even be used to control particles. (See Chapter 12, "Autodesk® Maya® Dynamics and Effects.") When used as a texture, a ramp can be connected to any attribute of a shader to create graduating color scales, transparency effects, increasing glow effects, and so on.

The *Ramp shader* is a self-contained shader node that automatically has several ramp texture nodes (a.k.a. gradients) attached to its attributes. You'll see what Ramp textures do later in the chapter. This makes for a simplified editing environment for the shader because all the colors and handles are accessible through the Ramp shader's own Attribute Editor, as shown in Figure 7.8.

To create a new color in any of the horizontal ramps, click in the swatch to create a new ramp position. Edit its color through its Selected Color swatch. You can move the

position by grabbing the circle right above the ramp and dragging left or right. To delete a color, click the box beneath it.

These ramp textures are automatically attached to the Color, Transparency, Incandescence, Specular Color, Reflectivity, and Environment attributes of a Ramp shader. In addition, a special curve ramp is attached to the Specular Roll Off attribute to give you more precise control over how the specular highlight diminishes over the surface.

Shaderfx Shader Type

This new shader sits at the top of the Create Panel in the Hypershade; it is an editing tool for creating shader networks for game engines. Because this is an advanced shader, its functions will not be covered in this book.

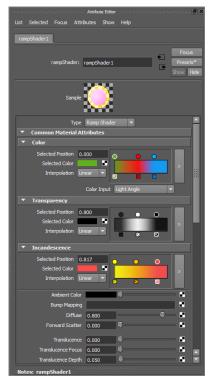
Shader Attributes

Shaders are composed of nodes just like other Maya objects. Within these nodes, attributes define what shaders do. Here is a brief rundown of the common shader attributes with which you'll be working:

Color An RGB or HSV value defines what color the shader is when it receives a neutral color light. For more on RGB and HSV, see the "RGB and HSV" sidebar in this chapter.

Figure 7.8

A Ramp shader in its
Attribute Editor



RGB AND HSV

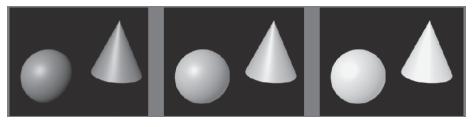
Computers represent color as sets of numeric values corresponding to the red, green, and blue channels of the image. An 8-bit image ranges from 0 to 255, so for each primary color you have 256 possible levels. With three channels, you have 256×256×256 (16.7 million) possible combinations of each primary color mixed to form the final color. Each RGB channel defines how much of that color is present in the displayed color.

Color value can also be set on the hue, saturation, and value (HSV) channels. The hue value defines the actual tint (from red to green to violet) of the color. The saturation defines how much of that tint is present in the color. Finally, value defines the brightness of the color, from black to white. The higher the value, the brighter the color will be.

Transparency The higher the Transparency value, the less opaque and more see-through the object becomes. Although usually expressed in a black-to-white gradient, with black being opaque or solid and white being totally clear, transparency can have color. In a color transparency, the shader's color shifts because only some of its RGB values are transparent, as opposed to the whole.

Ambient Color This color affects the overall color attribute of a shader because this adds a flat ambient color irrespective of lighting in the scene. The default is black, which keeps the darker areas of a surface dark. The lighter the ambient color, the lighter those areas become. A bright Ambient Color setting flattens out an object, as shown in Figure 7.9.

Figure 7.9 Ambient color values flatten an obiect.



Ambient Color = Black

Ambient Color = Medium Grav

Ambient Color = White

Incandescence This attribute is the ability to self-illuminate. Objects that seem to give off or have their own light, such as an office's fluorescent light fixture, can be given an Incandescence value. Incandescence doesn't, however, light objects around it in regular renders, nor does it create a glow. Also, it tends to flatten the object into a pure color (see Figure 7.10). As you'll see in Chapter 11, "Autodesk" Maya" Rendering," incandescence can also help light a scene in the mental ray[®] Final Gather rendering.

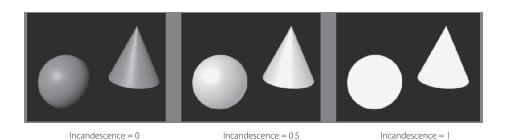


Figure 7.10
Incandescence values

Bump Mapping This attribute creates a textured feel for the surface by adding highlights and shadows to the render. It doesn't alter the surface of the geometry, although it makes the surface appear to have ridges, marks, scratches, and so forth. The more intense the variation in tones of a bump map, the greater the bump. Bump maps are frequently used to make surfaces look more real because nothing in reality has a perfectly smooth surface. Using bumps very close up may create problems; bumps are generally good for adding inexpensive detail to a model that isn't in extreme close-up (see Figure 7.11).

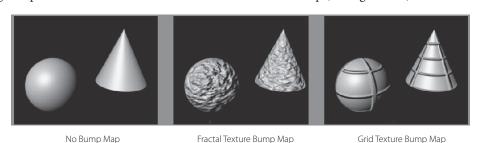
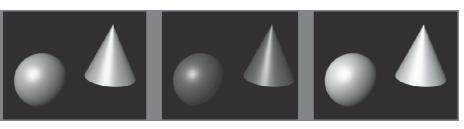


Figure 7.11
The effects of a bump map

Bump maps on geometry when seen close up don't work well. Those cases require displacement maps. I'll cover displacement maps in Chapter 11.

Diffuse This value governs how much light is reflected from the surface in all directions. When light strikes a surface, light disperses across the surface and helps illuminate it. The higher this value, the brighter its object is when lit. The lower the Diffuse value, the more light is "absorbed" into the surface, yielding a darker result (see Figure 7.12).



Diffuse = 0.8 (default) Diffuse = 0.3 Diffuse = 1

Figure 7.12

How a Diffuse value affects a shader's look

Translucence and Translucence Focus The Translucence and Translucence Focus attributes give the material the ability to transmit light through its surface, like a piece of canvas in front of a light. At a value of 1 for Translucence, all light shines through the object; at 0, none does. The Translucence Focus attribute specifies how much of that light is scattered. A light material such as paper should have a high translucence focus, and thicker surfaces should have low focus rates.

Glow Intensity Found in the Special Effects section of the Attribute Editor, the Glow Intensity attribute adds a glow to the object, as if it were emitting light into a foggy area (see Figure 7.13). You'll add glow to an object in Chapter 10, "Autodesk" Maya" Lighting,"

Figure 7.13 Adding a glow



Matte Opacity Objects rendered through Maya generate a solid *matte*. Where there is a solid object, the matte is white; where there is nothing, the matte is black. In compositing programs, which bring together elements created independently into a single composite scene, mattes help to separate rendered CG from their backgrounds. Turning down the slider decreases the brightness of the object's matte, making it appear more transparent. For more information about mattes, see the sidebar "Image Mattes" in this chapter, and see Chapter 11.

Raytrace Options With raytracing, you can achieve true reflections and refractions in your scene. This subset of attributes allows you to set the shader's raytracing abilities. See Chapter 11 for more on raytracing.

Some attributes are available only with certain shader types. The following are the attributes for the Phong, Phong E, and Blinn shaders because they can be shiny and reflective:

Specular Color This sets the color of the highlights on a shiny surface. Black produces no specular, and white creates a bright one.

Reflectivity This sets the amount of reflection visible in the surface. The higher the value, the more reflective the object will render. Increasing this value increases the visibility of the Reflected Color attribute or of true reflections in the scene when raytraced.

Reflected Color This gives the surface a reflection. Texture maps are generally assigned to this attribute to give the object a reflection of whatever is in the image file or texture without having to generate time-consuming true reflections with a raytraced render. Using

raytracing to get true reflections, however, is the only way to generate reflections of other objects in the scene.

Cosine Power This is available only with a Phong shader. This attribute changes the size of the shiny highlights (a.k.a. specular) on the surface. The higher the number, the smaller the highlight looks.

IMAGE MATTES

Image files are stored with a red, a green, and a blue channel that keep the amount of each color in each pixel of the image. Some image formats, including TIFF and TARGA, also have an alpha channel, known as a matte channel or image matte. This is a grayscale channel that controls the opacity of an image. Completely white parts of the matte make those parts of the image opaque (solid), whereas black parts make those parts of the image fully transparent. Gray in the matte channel makes those parts of the image partly transparent. These mattes are used in compositing—bringing together elements created separately into a single composite scene. See Chapter 11 for an example of how an alpha channel works.

Roughness, Highlight Size, Whiteness These control the specular highlight on a Phong E surface only. They control specular focus, amount of specular, and highlight color, respectively.

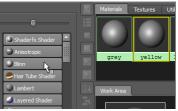
mental ray Attributes Because Autodesk integrates features of the Nvidia mental ray rendering engine into Maya, an object's Attribute Editor usually includes a set of mental ray options. Shaders are no different. If you open the Mental Ray heading in the Attribute Editor for a shader, you'll see attributes such as Reflection Blur and Irradiance, as well as a few ways to override the Maya shading attributes with those of mental ray. You'll work with mental ray in Chapters 10 and 11.

Shading and Texturing the Toy Plane

In the following section, you'll add shaders to the toy plane from Chapter 6, "Practical Experience!" This exercise is also available in a video on the book's web page, as well as the author's YouTube account. The plane is fairly straightforward to shade. Use bodyModel_ v06.ma from the plane project from the companion website. You have just two colors on the body of the plane. You'll start by making shaders in the Hypershade.

1. With either your scene file or bodyModel_v06.ma loaded, open the Hypershade (Window → Rendering Editors → Hypershade). You need to re-create a plastic look for the toy plane. Create two Blinn shaders by clicking Blinn in the Create column in the Hypershade twice, as shown in Figure 7.14. Name the first shader **gray** and the second shader **yellow**.

Figure 7.14 Create two Blinn shaders.



The toy plane is yellow and gray, but it has some flakes of dark gray in both colors. Figure 7.15 is a close-up photo of a yellow pontoon to showcase the dark "noise" in the plastic color. You will mimic that with a texture node called Noise.

Figure 7.15 A close-up of a yellow pontoon shows flakes of dark gray

in the plastic color.



2. In the Create Panel of the Hypershade, click 2D Textures to display the 2D texture nodes. Then click Noise to create a new noise texture, as shown in Figure 7.16.

Figure 7.16 Click to create a new noise texture node.

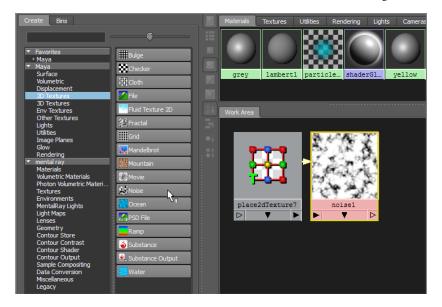


Table 7.1 Noise texture values

ATTRIBUTE	VALUE		
Threshold	0.725		
rnresnoia	0.735		
Amplitude	0.56		
Ratio	2.02		
Frequency Ratio	6.62		
Spottyness	1.4		

3. It doesn't look like much now, but you will adjust the attributes on this noise texture to simulate the dark flakes in the plastic. With the noise texture node still selected, in the Attribute Editor set the attribute values as shown in Table 7.1, leaving the remaining attributes at their default values.

Also in the Color Balance heading, click the white color swatch next to the Color Gain attribute, and set that color to a mustard yellow using the HSV values of 46.15,

- **0.51**, **0.60**, as shown in Figure 7.17. Call the node **noiseYellow**. This color will give you the yellow you need for that plastic.
- 4. You'll duplicate the yellow noise node to make a gray version for the gray plastic parts of the plane. In the Hypershade, select noiseYellow and choose Edit → Duplicate → Shading Network. Name the duplicate texture node noiseGray.
- 5. In the Color Balance heading for the noiseGray node, set the Color Gain HSV values to 231.5, 0.23, 0.39.
- 6. Select the grey shader in the Hypershade to display it in the Attribute Editor. MMB+drag the grayNoise node to the Color attribute of the gray shader in the Attribute Editor, as shown in Figure 7.18. This will set the gray noise texture you just made to be the color of the gray plastic material.
- 7. Select the yellow Blinn shader and MMB+drag the noiseYellow node to the Color attribute to connect the yellow color to the yellow Blinn shader.

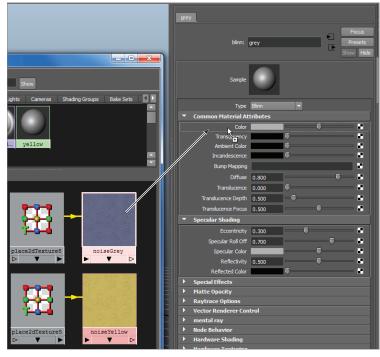


Figure 7.18

Connect the gray noise texture to the Color attribute of the gray Blinn shader.

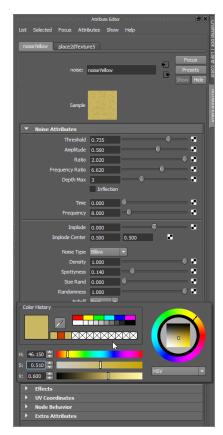
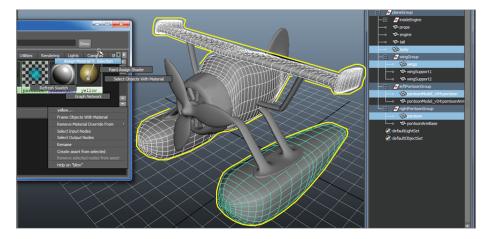


Figure 7.17 Setting the noise texture values

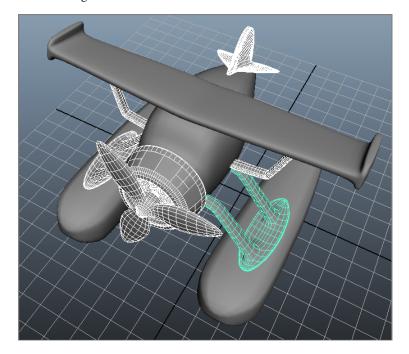
8. Now let's assign the shaders to the parts of the plane. In the persp view panel, select the wings, body, and the two pontoons and assign the yellow Blinn shader to them by RMB+clicking in the Hypershade and selecting Assign Material To Selection, as shown in Figure 7.19.

Figure 7.19 Select these parts of the plane and assign the yellow Blinn to them.



9. Select the remaining parts of the plane and assign the gray Blinn shader to them, as shown in Figure 7.20.

Figure 7.20 Select the remaining parts of the plane and assign the gray Blinn shader to them.



Now let's see what it looks like! In the persp view panel, render a view of your plane by clicking the Render Current frame icon on the Status line (). Since you haven't created any lights in the scene, Maya will use default lights and render an image (Figure 7.21) with the basic shaders you just created. Not bad. Feel free to adjust the colors or the attributes of the noise textures to your liking. Once you start lighting the scene in Chapter 10, you will adjust the shaders to make the toy plane look like better plastic. For now, you're done! You can check your work against the scene file planeShading_v01.ma in the scenes folder of the Plane project.

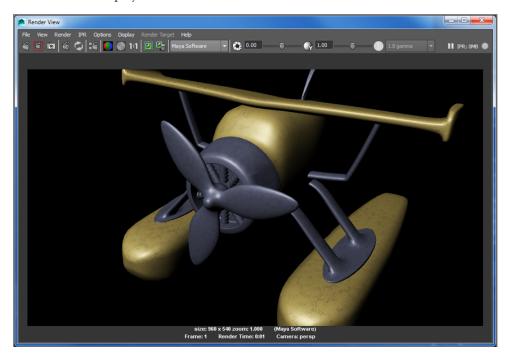


Figure 7.21

The plane renders with yellow and gray Blinn shaders to look a little like plastic.

You can embellish a model a lot at the texturing level. The more you explore and experience shaders and modeling, the better you'll be at juggling modeling with texturing to get the most effective solution. For example, you can use more elaborate noise patterns to mimic the flakes in the plane's plastic, as opposed to using the simple pattern you created with a noise texture. That is a more involved workflow, which you will tackle later in this chapter with UVs and texture maps.

You'll begin UV layout and texturing a child's red wagon later in this chapter and then go into more detailed texturing with the decorative box, which you'll then light and render in mental ray. For even more practice, try loading the catapult model from Chapter 4, "Beginning Polygonal Modeling," and texturing it from top to bottom.

Textures and Surfaces

Texture nodes generate maps to connect to an attribute of a shader. There are two types of textures: procedural and bitmapped (sometimes called maps). Procedural textures use Maya attributes to generate an effect, such as a ramp gradient, checkerboard, or fractal noise texture. You can adjust each of these procedural textures by changing their attribute values.

A map, on the other hand, is a saved image file that is imported into the scene through a file texture node. These files are pregenerated through whatever imaging programs you have and include digital pictures and scanned photos. You need to place all texture nodes onto their surfaces through the shader. You can map them directly onto the UV values of the surface or project them.

UV Mapping

UV mapping places the texture directly on the surface and uses the surface coordinates (called UVs) for its positioning. In this case, you must do a lot of work to line up the UVs on the surface to make sure the created images line up properly. What follows is a brief summary of how UV mapping works. You'll get hands-on experience with UV layout with the red wagon and decorative box exercises later in this chapter.

Just as 3D space is based on coordinates in XYZ, surfaces have coordinates denoted by U and V values along a 2D coordinate system for width and height. The UV value helps a texture position itself on the surface. The U and V values range from 0 to 1, with (0,0) UV being the origin point of the surface.

Maya creates UVs for primitive surfaces automatically, but frequently you need to edit UVs for proper texture placement, particularly on polygonal meshes after you've edited them. In some instances, placing textures on a poly mesh requires projecting the textures onto the mesh because the poly UVs may not line up as expected after the mesh has been edited. See the next section, "Using Projections."

If the placement of your texture or image isn't quite right, simply use the 2D placement node of the texture node to position it properly. See the section "Texture Nodes" later in this chapter for more information.

Using Projections

You need to place textures on the surface. You can often do so using UV placement, but some textures need to be projected onto the surface. A projection is what it sounds like. The file image, ramp, or other texture being used can be *beamed* onto the object in several ways.

You can create any texture node as either a normal UV map or a projected texture. In the Create Render Node window, clicking a texture icon creates it as a normal mapped texture. To create the texture as a projection, you must RMB+click the icon and select Create As Projection (see Figure 7.22).

When you create a projected texture, a new node is attached to the texture node. This projection node controls the method of projection with an attached 3D placement node, which you saw in the axe exercise. Select the projection node to set the type of projection in the Attribute Editor (see Figure 7.23).



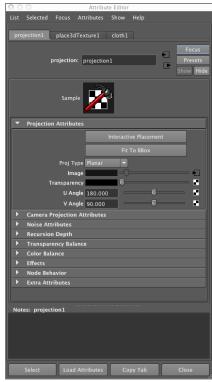


Figure 7.22 Selecting the type of map layout

Figure 7.23 The projection node in the Attribute Editor

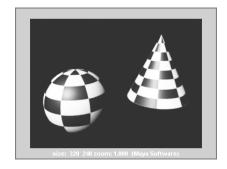
Setting the projection type will allow you to project an image or a texture without having it warp and distort, depending on the model you're mapping. For example, a planar projection on a sphere will warp the edges of the image as they stretch into infinity on the sides of the sphere.

Try This In a new scene, create a NURBS sphere and a NURBS cone and place them side-by-side. Create a Blinn shader and assign it to both objects. In the Blinn shader's Attribute Editor, set its Color attribute to a checkerboard pattern, as shown in Figure 7.24.

Try removing the color map from the Blinn shader. In the Blinn shader's Attribute Editor, RMB+click and hold the attribute's title word Color; then choose Break Connection from the context menu. Doing so severs the connection to the checker texture map node and resets the

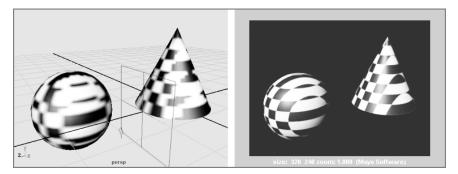
Figure 7.24 Assign a checkerboard pattern to the sphere and cone with Normal

checked.



color to gray. Now, re-create a new checker texture map for the color, but this time create it as a projection by RMB+clicking the icon in the Create Render Node window. In the illustration on the left in Figure 7.25, you see the perspective view in Texture mode (press the 6 key) with the two objects and the planar projection placement node.

Figure 7.25 A planar projection checkerboard in the view panel (left) and rendered (right)



Try moving the planar placement object around in the scene to see how the texture maps itself to the objects. The illustration on the right of Figure 7.25 shows the rendered objects.

Try the other projection types to see how they affect the texture being mapped.

Projection placement nodes control how the projection maps its image or texture onto the surface. Using a NURBS sphere with a spherical projected checker, with U and V wrap turned off on the checkerboard texture, you can see how manipulating the place3dTexture node affects the texture.

In addition to the Move, Rotate, and Scale tools, you can use the special manipulator (press T to activate or click the Show Manipulator Tool icon in the toolbar) to adjust the placement. Figure 7.26 shows this tool for a spherical projection.

Drag the handles on the special manipulator to change the coverage of the projection, orientation, size, and so forth. All projection types have special manipulators. Figure 7.27 shows the manipulator wrapping the checkerboard pattern in a thin band all the way around the sphere.

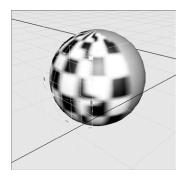


Figure 7.26 The spherical projection's manipulator tool

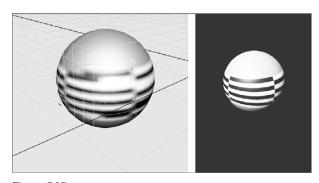


Figure 7.27 The manipulator wrapping the checkerboard texture around the sphere (left, perspective view; right, rendered view)

To summarize, projection textures depend on a projector node to position the texture onto the geometry. If the object assigned to a projected texture is moving, consider grouping the projection placement node to the object itself in the Outliner.

Texture Nodes

You can create a number of texture nodes in Maya. This section covers the most important. All texture nodes, however, have common attributes that affect their final look. Open the Attribute Editor for any texture node (see Figure 7.28). The Color Balance and Effects headings affect the color of the texture (as you saw with the noise textures from the plane exercise).

Color Balance This set of attributes adjusts the overall brightness and color balance of your texture. Use these attributes to tint or brighten a texture without having to change all the individual attributes of the shader.

Effects You can invert the texture's color space by clicking the Invert check box. This changes black to white and white to black in addition to inverting the RGB values of colors.

You can map textures to almost any shader attribute for detail. Even the tiniest amount of texture on a surface's bump, specular, or color increases its realism.

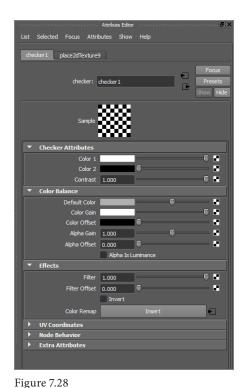
Place2dTexture Nodes

The 2D texture nodes come with a 2D placement node that controls their repetition, rotation, size, offset, and so on. Adjust the setting in this node of your 2D texture in the Attribute Editor, as shown in Figure 7.29, to position it within the Shader network.

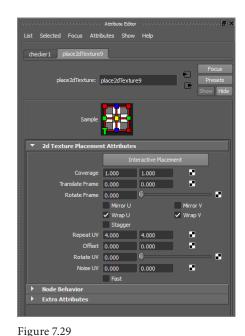
The Repeat UV setting controls how many times the texture is repeated on whatever shader attribute it's connected to, such as Color. The higher the wrap values, the smaller the texture appears, but the more times it appears on the surface.

The Wrap U and Wrap V check boxes allow the texture to wrap around the edges of their limits to repeat. When these check boxes are turned off, the texture appears only once, and the rest of the surface is the color of the Default Color attribute found in the texture node.

The Mirror U and Mirror V settings allow the texture to mirror itself when it repeats. The Coverage, Translate Frame, and Rotate



Some common attributes for all texture nodes



A 2D placement node in the Attribute Editor

Frame settings control where the image is mapped. They're useful for positioning a digital image or a scanned picture.

Ramp Texture

A *ramp* is a gradient in which one color transitions into the next color. A ramp is useful for many things, but mostly it's perfect for making color gradients. Initially the ramp texture will show as a simple blackto-white gradient (from left to right), as shown in Figure 7.30.

But it's easy to create gradients using any colors. Simply use the round handles on top of the ramp to select that color and to move it left and right on the ramp. The square handle at the bottom deletes the color. To create a new color, click inside the ramp. Figure 7.31 shows a multicolored ramp. Clicking the right arrow button to the right of the ramp shown in Figure 7.31 will open a larger ramp window to work with if you need extra precision.

The ramp texture is different from the Ramp shader. The Ramp shader automatically has several ramp textures mapped to some of its attributes.

The Type setting allows you to create a gradient running along the U or V direction of the surface, as well as to make circular, radial, diagonal, and other types of gradients. The Interpolation setting controls how the colors grade from one to the next. Experiment with them to see how they work.

The U Wave and V Wave attributes let you add a squiggle to the U or V coordinate of the ramp, and the Noise and Noise Freq (frequency) attributes specify randomness for the placement of the ramp colors throughout the surface.

Using the HSV Color Noise attributes, you can specify random noise patterns of hue, saturation, and value to add some interest to your texture. The HSV Color Noise options are great for making your shader just a bit different to enhance its look.

Fractal, Noise, and Mountain Textures

These textures are used to create a random noise pattern to add to an object's color, transparency, or any other shader attribute. For example, when creating a surface, you'll almost always want to add a little dirt or a few surface blemishes to the shader to make the object look less CG. These textures are commonly used for creating bump maps.

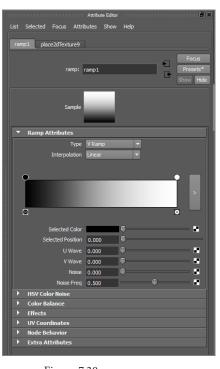


Figure 7.30 The ramp texture



Figure 7.31 Adding and changing colors in a ramp is easy!

These textures help create surface features when used on a shader's Bump Mapping attribute. Each creates an interesting pattern to add to a surface to create tactile detail, but you can also use them to create color or specular irregularities.

When used as a texture for a bump, Grid is useful for creating the spacing between tiles, Cloth is perfect for clothing, and Checker is good for rubber grips. Placing a Water texture on a slight reflection makes for a nice poolside reflection in patio furniture.

The File Node

You use the file node to import image files into Maya for texturing. For instance, if you want to texture a CG face with a digital picture of your own face, you can use the file node to import a Maya-supported image file.

Importing an Image File as a Texture

To attach an image to the Color attribute of a Lambert shader, for example, follow these steps:

- 1. Create a Lambert shader in the Hypershade. (Phong, Blinn, or any of the shaders will do.)
- 2. Click the Map button () next to the Color attribute of the new Lambert shader to map a texture to it. RMB+click the File button in the Create Render Node window that pops up and select Create Texture. (You can also click to create the image file as a projected texture if you want.) The Attribute Editor shows the attributes for the file node; see Figure 7.32.
- 3. Next to the Image Name attribute, click the Folder icon to open the file browser. Find the image file of choice on your computer. (It's best to put images to use as textures in the project's Sourceimages folder. As a matter of fact, the file browser defaults directly to the Sourceimages folder of your current project.) Double-click the file to load it.
- 4. After you import the image file, it connects to the Color attribute of that shader and also automatically connects the alpha to transparency if there is an alpha channel in the image. You can position it as you please by using its place2dTexture node or by manipulating the projection node if you created the file texture node as a projection.

You can attach an image file to any attribute of a shader that is *mappable*, meaning it's able to accept a texture node. Frequently,

Figure 7.32
The file texture node



image files are used for the color of a shader as well as for bump and transparency maps. You can replace the image file by double-clicking the file texture node in the Hypershade and choosing another image file with the file browser.

Using Photoshop Files: The PSD File Node

Maya can also use Adobe Photoshop PSD files as image files in creating shading networks. The advantage of using PSD files is that you can specify the layers within the Photoshop file for different attributes of the shader, as opposed to importing several image files to map onto each shader attribute separately. This, of course, requires a modest knowledge of Photoshop and some experience with Maya shading. As you learn how to shade with Maya, you'll come to appreciate the enhancements inherent in using Photoshop networks.

Try This You'll create a single Photoshop file that will shade this sphere with color as well as transparency and a bump. Again, you're doing this instead of creating three different image files (such as TIFFs) for each of those shading attributes.

1. Create a NURBS sphere in a new scene and assign a new Lambert shader to it. You can do this through the Hypershade or by choosing Lighting/Shading → Assign Favorite Material → Lambert while in the Rendering menu set. This creates a new shader and assigns it to the selection, in this case your sphere.

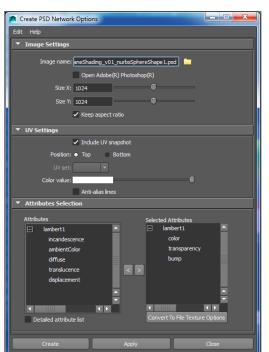


Figure 7.33 The Create PSD Network Options window

- 2. Select the sphere and choose Texturing → Create PSD Network. In the option box that opens, select Color, Transparency, and Bump from the list of attributes on the left side and click the right arrow to move them to the Selected Attributes list on the right, as shown in Figure 7.33.
- 3. Select a location and filename for the image. By default, Maya places the PSD file it generates in the Sourceimages folder of your current project, named after the surface to which it applies. Click Create.
- 4. In Photoshop, open the newly created PSD file. You see three layers grouped under three folders named after the shader attributes you selected when creating the PSD file. There are folders for lambert2.bump, lambert2. transparency, and lambert2.color, as well as a layer called UVSnapShot.

The UVSnapShot layer gives you a wireframe layout of the UVs on the sphere as a guideline to paint your textures. Because the sphere is an easy model, you don't need this layer, so turn it off. You'll use UV SnapShot later in this chapter when texturing a toy wagon.

You can now paint whatever image you want into each of the layers to create maps for each of the shader attributes, all in one convenient file. Save the PSD file. You can save over it or create a new filename for the painted file.

- 5. In Maya, open the Hypershade, and open the Attribute Editor for the Lambert shader assigned to the sphere (in this case, Lambert2). If you graph the connections to the Lambert shader in the Hypershade window's work panel, you see that the PSD file you generated is already connected to the Color, Bump, and Transparency attributes of the shader, with the proper layering set for you, as shown in Figure 7.34.
- 6. Open the file nodes for the shader and replace the PSD file with your new painted PSD file. If you saved your PSD file with the same name, all you need to do is click the Reload button to update the psdFile node.

If you decide you need to add another attribute to the PSD file's layering or if you need to remove an attribute, you can edit the PSD network. Select the shader in the Hypershade and choose Edit -> Edit PSD Network. In the option box, select new attributes to assign to the PSD file, or remove existing attributes and their corresponding Photoshop layer groups. When you click Apply or Edit, Maya saves over the PSD file with the new layout.

3D and Environment Textures

3D textures are projected within a 3D space. These textures are great for objects that need to reflect an environment, for example.

Instead of simply applying the texture to the plane of the surface as 2D textures do, 3D textures create an area in which the shader is affected. As an object moves through a scene with a 3D placement node, its shader looks as if it swims, unless that placement node is parented or constrained to that object. (For more on constraints, see Chapter 9, "More Animation!")

Disconnecting a Texture

Sometimes, the texture you've applied to an object isn't what you want, and you need to remove it from the shader. To do so, double-click the shader in the Hypershade to open its Attribute Editor.

You can then disconnect an image file or any other texture node from the shader's attribute by RMB+clicking the attribute's name in the Attribute Editor and choosing Break Connection from the context menu, as shown in Figure 7.35.

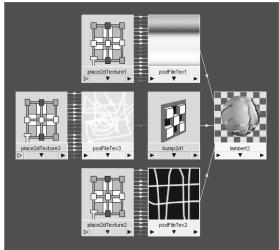
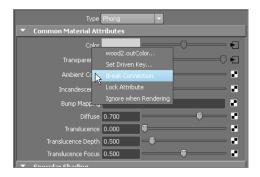


Figure 7.34

The PSD network for the Lambert shader in the Hypershade shows the connections to Color, Bump, and Transparency.

Figure 7.35 RMB+clicking a shader's attribute allows you to disconnect a texture node from the shader.



Textures and UVs for the Red Wagon

Now you'll use a child's red wagon toy model, to which you will assign shaders. The real wagon is shown in Figure 7.36. If you are reading a digital copy of this book, you can see the colors in Figure 7.36 easily. Otherwise, you can download the ColorWagon.tif photo from the book's website. The wagon is fairly simple; it will need a few colored shaders (Red, Black, Blue, and White) for the body, along with a few texture maps for the decals which is where the real fun begins. The wagon will also require some more intricate work on the shaders and textures for the wood railings and silver metal screws, bolts, and handlebar; these will be a good foray into image maps and UVs.

Figure 7.36 The red wagon and its named parts



This exercise is a prime example of how lighting and shading go hand in hand.

Assigning Shaders

Load the file RedWagonModel.ma from the Scenes folder of the RedWagon project to begin shading the model of the wagon.

Shading is the common term for adding shaders to an object.

Study the color images of the wagon and see how light reflects off its plastic, metal, and wood surfaces. Blinn shaders will be perfect for nearly all the parts of the wagon. Follow these steps:

- 1. Open the Hypershade window and create four Blinn shaders.
- 2. Assign the following HSV values to the Color attribute of each Blinn shader and name them as shown in Table 7.2 and in Figure 7.37. You'll create the Chrome Metal and Wood shaders later.

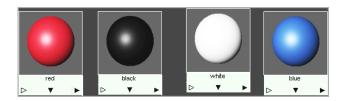


Figure 7.37 Create the four colored Blinn shaders.

WAGON COLOR	SHADER NAME	H VALUE	S VALUE	V VALUE
Red	Red	355	0.910	0.650
Black	Black	0	0	0
White	White	0	0	1
Blue	Blue	220	0.775	0.560

Table 7.2 HSV color values for the wagon's colors

Initial Assignments

Look at the color image of the wagon from the book's web page (or in the e-edition of the book). The bullnose and tires are black, the wheel rims are white, the floor is blue, the screws and bolts and handlebar are chrome metal, the railings are wood, and the main body is red. Assign shaders to the wagon according to the color photo and the following steps:

- 1. In the view panel, select the side panels (the A and B panels, without the screws and bolts) and the wheel rim caps, as shown in Figure 7.38, and assign the Red Blinn shader to them. Press 6 to enter Texture Display mode.
- 2. Select the wheelMesh objects for all four wheels and assign them the White shader. The tires also turn white, but you'll fix that shortly; don't worry about it now. See Figure 7.39.

Figure 7.38

Assign the Red shader.

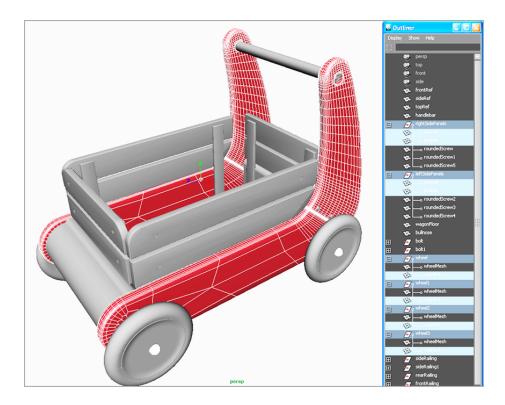
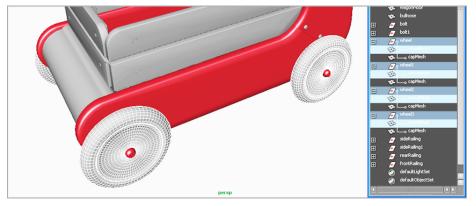


Figure 7.39

Assign the White shader.



- 3. Select the bullnose (the rounded cylinder in front of the wagon), and assign it the Black Blinn.
- 4. Select the wagonFloor object and assign it the Red shader, as shown in Figure 7.40. You'll notice that the front and back bodies of the wagon turn red like they're supposed to, but so does the floor of the wagon, which should be blue according to the photo. If you try to assign the Blue shader to the wagon floor mesh, the floor will be correct, but the front and back of the wagon will be blue and not red. You'll fix this later.

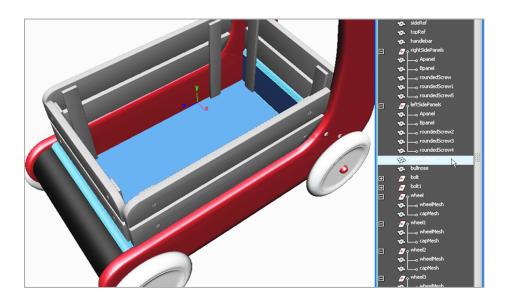


Figure 7.40 Assign the Red shader to the wagon floor just for now;

you'll change it to blue later.

Now you have initial assignments for the basic colors of the wagon's body. Let's tweak these shaders' colors next.

Creating a Shading Network for the Wheels

Refer to Figure 7.41 to observe how the materials are different for the rim and the tire of the wheel. The rim is glossier and has a tighter, sharper specular, whereas the tire has a diffuse specular and is quite bumpy. You'll create a Layered shader for the wheels with white feeding into the rim portion and black into the tire portion.

Coloring the Wheel

First you need to determine where the white ends and the black starts on the surface of the wheel mesh.

1. Select the White shader in the Hypershade window. Click the Map icon (▶) next to the Color attribute. RMB+click the Ramp Texture icon and select Create Texture. The wheel's color turns to a blackto-white gradient clockwise around the wheel, which is the wrong direction; you need the gradient to run from the center to the edge. In the ramp texture's Attribute Editor, set Type to U Ramp, as shown in Figure 7.42.

If the ramp doesn't show up in your view panels, make sure you press 6 to enter Texture Display mode. If the colors and ramp texture still don't display, make sure Use Default Material isn't checked in the view panel's Shading menu.

Figure 7.41 The tire on the wagon



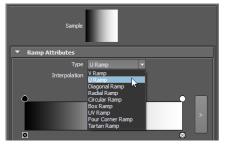
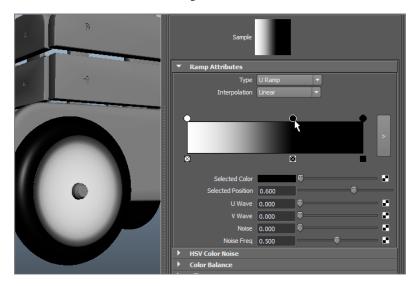


Figure 7.42 Set the ramp to a U Ramp type.

- 2. Now the gradient is running black from the center to light gray on the outside edge through to white on the reverse side of the wheel. Select the round handle for the black color and change Selected Color to white. Then select the round white handle on the right side of the ramp and change Selected Color to black.
- 3. Click in the middle of the ramp to create a new color, which will be gray. Set that color to black. Move that new black handle in the ramp until its Selected Position value is about 0.6, as shown in Figure 7.43.

Figure 7.43 Move the new black handle.



- 4. Set the Interpolation attribute (found above the ramp color) to None so you get clean transitions from white to black, instead of a soft linear gradient where the black slowly grades to white. Name this ramp wheelPositionRamp.
- 5. The backs of the wheels are solid black. In the wheelPositionRamp, move the round handle for the black color all the way at the right of the ramp and set that color to white; set its position to about 0.920 to place white behind the wheels, keeping the black only where the tire is. See Figure 7.44.

Now that you've pinpointed where the white rim ends and the black tire begins, you'll use this ramp as a transparency texture to place the Tire shader on top of the Rim shader in a Layered shader that you'll create later. This will allow you to have two different materials on your tire, a smooth white plastic and a rough black plastic.

6. You don't need this Ramp shader on the color of the shader anymore—you did this so you could easily see the ramp color positions in the view panels. In the Hypershade, select the White shader and click the Input And Output Connections icon () to graph the shader, as shown in Figure 7.45.

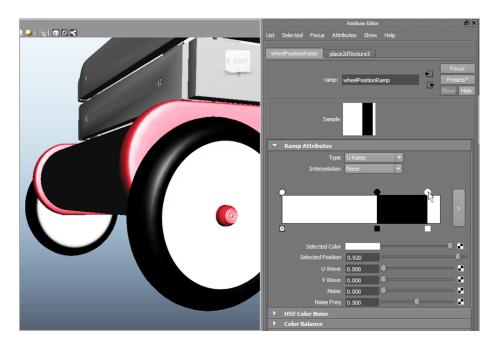


Figure 7.44 Setting the tire location using a ramp

7. In the Attribute Editor, RMB+click the Color attribute and select Break Connection from the context menu to disconnect the ramp from the color. Set the Color back to white. Notice that the rubber band link connecting the ramp texture node to the White shader node disappears in the Hypershade window.

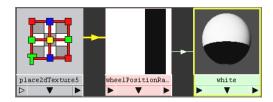
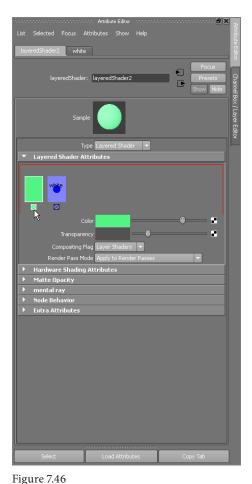


Figure 7.45

The White shader has the ramp attached as color.

- 8. Create a Layered shader. MMB+drag the White shader from the Hypershade to the top of the Layered Shader Attributes window, as shown in Figure 7.46. Delete the default Green shader in the Attribute Editor by clicking the checked box below its swatch.
- 9. Create a new Blinn shader and set its color to black. Name the shader tireShader. Select the Layered shader and then MMB+drag the new tireShader from the Hypershade to the Layered shader's Attribute Editor, placing it to the left of the White shader, as shown in Figure 7.47. Name the Layered shader wheelShader.
- 10. Select the wheels and assign the wheelShader Layered shader to them. The wheels should appear all white. This is where the ramp you created earlier (wheelPosition-Ramp) comes into play.





Drag the White shader to the Layered shader and delete the default Green shader from it.

Figure 7.47 Add the black tireShader.

- 11. Select the wheelShader and click Input And Output Connections to graph the network in the Work Area of the Hypershade window. In the top panel of the Hypershade, click the Texture tab to display the texture nodes in the scene so you can see wheelPositionRamp's node.
- 12. In the wheelShader's Attribute Editor, click the tireShader swatch on the left. MMB+drag the wheelPositionRamp node to the Transparency attribute, as shown in Figure 7.48.
- 13. When you attach the ramp, the wheelShader icon turns white on top and black on the bottom. Render a frame in the persp view panel to make sure the black tires line up properly, as shown in Figure 7.49. The wheel coloring is complete!

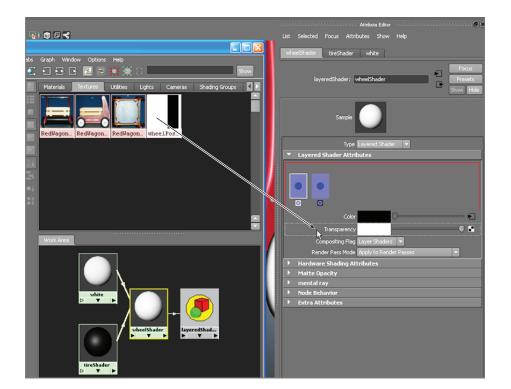


Figure 7.48

Attach the ramp to the Transparency attribute of the tireShader in the wheelShader.



Figure 7.49

The tires are done!

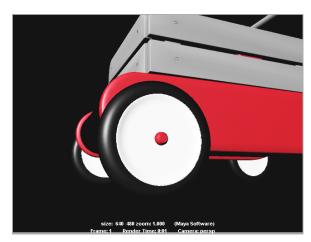
HEY, HOLD ON A MINUTE: WHY TWO SHADERS?

Why did you go through a Layered shader with two different shaders (one white and one black) when you could more easily use one shader and assign the same black-to-white ramp to its color? You did this because the white rim and the black tire are different materials, and you need to use two different shaders to properly show that in renders.

Setting the Feel for the Materials and Adding a Bump Map

Because the material look and feel on the real wheels differs quite a bit between the rim and the tire, you'll further tweak the white rim and the black tire shaders. The rim is a smooth, glossy white, and the tire is a bumpy black with a broad specular. Follow these steps:

- 1. Let's set up a good angle of view for your test renders. Position your persp view panel to resemble the view in Figure 7.50. Render a frame.
- 2. Select the White shader and set Eccentricity to **0.05** and Specular Roll Off to **0.5**. Doing so sharpens the specular highlight on the rim.
- 3. Select the black tireShader and set Eccentricity to 0.375 and Specular Roll Off to 0.6 to make the highlight more diffuse across the tire. Set Reflectivity to **0.05**. Render a frame; refer to Figure 7.51 to see how the specular highlight is broader and less glossy than in Figure 7.50.
- 4. Open the Attribute Editor for the tireShader, and click the Map icon (■) next to the Bump Mapping attribute. In the Create Render Node window, click to create a fractal texture map. Notice that the entire wheelShader icon becomes bumpy. Render a frame, and you'll see that the entire wheel is bumpy—not just the tire. (See Figure 7.52.) Argh!





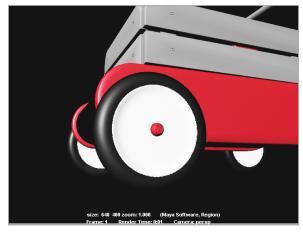


Figure 7.51 Setting specular levels



Figure 7.52 The whole wheel becomes bumpy!

5. You have to use the wheelPositionRamp to prevent the bump from showing on the rim. Select the wheelShader and click the Input And Output Connections icon () in the Hypershade to graph its network (Figure 7.53).

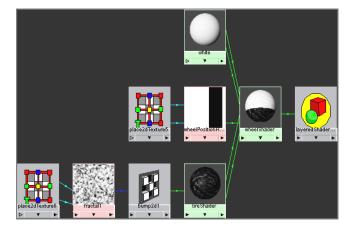
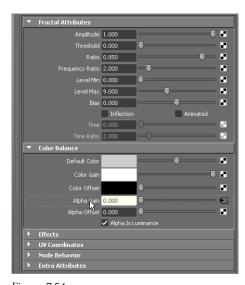


Figure 7.53 Graph the wheel-Shader network.

Notice the single blue connecting line between the fractal1 node and the bump2d1 node in the Hypershade. This means the alpha channel of the fractal feeds the amount of bump that is rendered on the tireShader. You have to alter the alpha coming out of the fractal node with the positioning ramp to block the rim from having any bump. The white areas of the ramp allow the bump map from the fractal to show on the tire, whereas the black area of the ramp keeps any bump from appearing. Because you already used this ramp to position the Rim shader and the Tire shader on the wheel, it will work perfectly for the bump position as well.

- 6. You have to use the wheelPositionRamp to prevent the bump from showing on the rim. Select the wheelShader and click the Input And Output Connections icon () in the Hypershade to graph its network (Figure 7.53).
- 7. Select the fractal to display it in the Attribute Editor and MMB+drag the wheelPositionRamp in the Hypershade to the Alpha Gain attribute for the fractal, as shown in Figure 7.54.
- 8. Render a frame, and you see that now the tire has no bump and the rim is bumpy (Figure 7.55).
- 9. This is easy enough to fix. All you need to do is reverse the ramp and then feed it into the Alpha Gain attribute of the fractal so that the tire is bumpy and the rim is smooth. In the Hypershade, in the Create pane on the left, click the Maya → Utilities heading and then click the Reverse icon to create a reverse node in the Hypershade window. See Figure 7.56.
- 10. In the Hypershade, MMB+drag the wheelPositionRamp node onto the Reverse node and select Input from the context menu when you release the mouse button. This connects the output of the ramp into the reverse node, which will then reverse the effect of the ramp on the fractal when you connect it in the next step.







The tire is smooth, and now the rim is bumpy.

- 11. MMB+drag the reverse node on top of the fractal 1 node and select Other from the context menu. This opens the Connection Editor, which you first saw in Chapter 3. On the left the reverse1 node is loaded, and on the right is the fractal1 node. In the
 - left pane, click the plus sign next to the Output attribute and select outputX. In the right pane, select the alphaGain attribute, as shown in Figure 7.57.
- 12. Open the Render Settings window by choosing Window → Rendering Editors → Render Settings. Click the Maya Software tab and set Quality to Production Quality in the pulldown menu. (See Figure 7.58.) Render a frame: you finally have a bump on the tire and a smooth rim. See Figure 7.59.
- 13. It's not a very convincing bump yet, so select the fractal node, and set Ratio to 0.85. Click the Placement tab for the fractal (it should be called something like *place2dTexture6*), and set the Repeat UV values to 18 and 48, as shown in Figure 7.60. Doing so makes the fractal pattern finely speckled on the tire.



Figure 7.56 Create a reverse node.

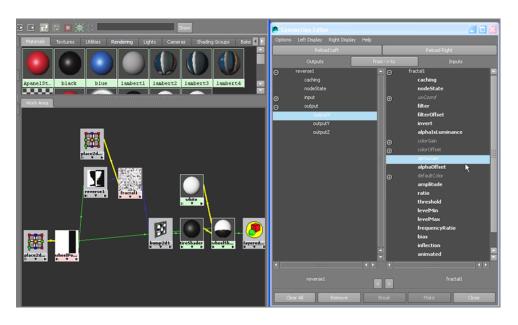
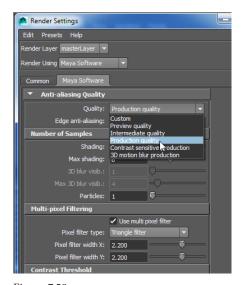


Figure 7.57

Connecting the reverse node to the fractal node



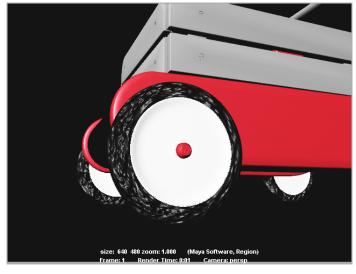


Figure 7.58

Set Quality to Production Quality.

Figure 7.59

Now you have the bump where you need it.

14. Render a frame; the fractal's scale on the bumpy tire looks too strong. Double-click the bump2d node in the Hypershade, and in the Attribute Editor, set Bump Depth to **0.04**. Render and check your frame against Figure 7.61. The bump looks much better, if not a little strong from this angle; you can finesse it to taste from here.

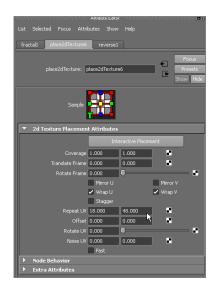


Figure 7.60

Set the Repeat UV values for the fractal map.

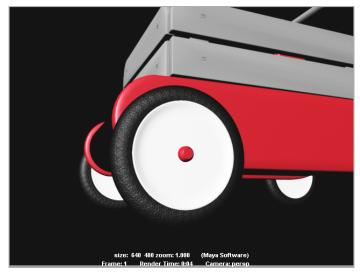


Figure 7.61

The wheel looks pretty good.

Tire Summary

Congratulations! You've made your first somewhat complex shading network, as shown in Figure 7.62. By now, you should have a pretty good idea of how to get around the Hypershade and create shading networks. To recap, you're using a ramp to place the two Tire and Rim shaders on the wheel, as well as using it to place the bump map on just the tire by using a reverse node. The more you make these shading networks, the easier they will become to create.

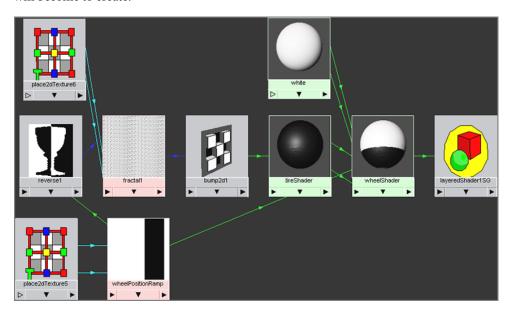


Figure 7.62 Your first complex shading network

This type of shading is called *procedural shading* because you used nothing but stock Maya texture nodes to accomplish what you needed for the wheels. In the following sections, you'll make good use of image mapping to create the decals for the wagon body as well as the wood for the railings.

You can load the file RedWagonTexture_v01.ma from the Scenes folder of the RedWagon project to check your work or skip to this point.

Putting Decals on the Body

Figure 7.63 shows you the decals that need to go onto the body of the wagon. They include the wagon's logo, which you'll replace with your own graphic design, and the white stripe that lines the side panels.

Instead of trying to make a procedural texture as you did with the wheel, you'll create an image map that will texture the side panels' white stripe. The stripe is far too

Figure 7.63 You need to add the body decals.

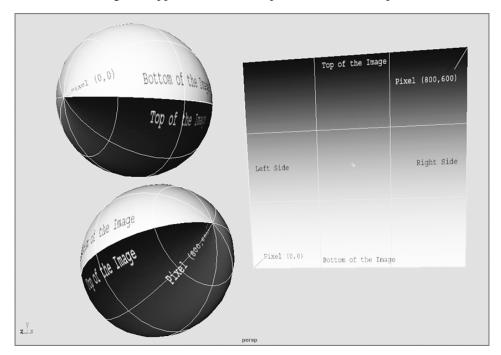


difficult to create otherwise. You'll create an image file using Photoshop (or another image editor) to make sure the white stripe (and later the red wagon logo) lines up correctly.

Working with UVs

Mapping polygons can involve the task of defining UV coordinates for them so that you can more easily paint an image map for the mesh. When you create a NURBS surface, UV coordinates are inherent to the surface. At the *origin* (or the beginning) of the surface, the UV coordinate is (0,0). When the surface extends all the way to the left and all the way up, the UV coordinate is (1,1). When you paint an 800×600-pixel image in Photoshop, for example, it's safe to assume that the first pixel of the image (at X = 0 and Y = 0 in Photoshop) will map directly to the UV coordinate (0,0) on the NURBS surface, whereas the top-right pixel in the image will map to the UV (1,1) of the surface. Toward that end, mapping an image to a NURBS surface is fairly straightforward. The bottom of the image will map to the bottom of the surface, the top to the top, and so on. Figure 7.64 shows how an image is mapped onto a NURBS plane and a NURBS sphere.

Figure 7.64 An image file is mapped to a NURBS plane and a NURBS sphere. Notice the locations marked in the image and how they map to the locations on the surface, with the pixel coordinates directly corresponding to the surface's UV coordinates.



The locations in the image, marked by text, correspond to the positions on the NURBS plane. The sphere, because it's a surface bent around spherically, shows that the origin of the UV coordinates is at the sphere's pole on the left and that the image wraps itself around it (bowing out in the middle) to meet at the seam along the front edge as shown.

When you're creating polygons, however, this isn't always the case. You must sometimes create your own UV coordinates on a polygonal surface to get a clean layout on which to paint in Photoshop. Although poly UV mapping becomes fairly involved and complicated, it's a concept that is important to grasp early. When poly models are created, they have UV coordinates; however, these coordinates may not be laid out in the best way for texture-image manipulation.

Working with the A Panels

Let's look at how the UVs are laid out for the A panel of the wagon model. Use Figure 7.36 earlier in the chapter to remind yourself where the A panel is.

1. Using the scene file RedWagonTexture_v01.ma from the Scenes folder of the RedWagon project, select the A panel on one side of the wagon and choose Window → UV Texture Editor, as shown in Figure 7.65.

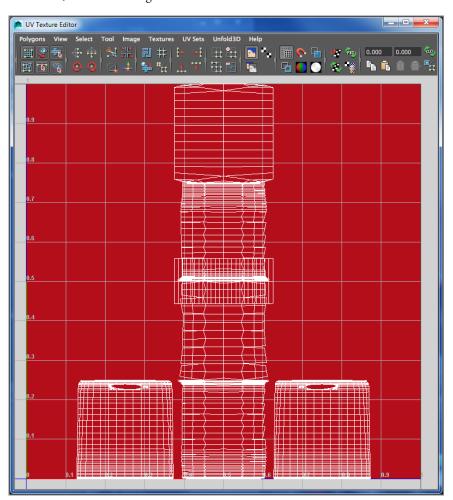


Figure 7.65

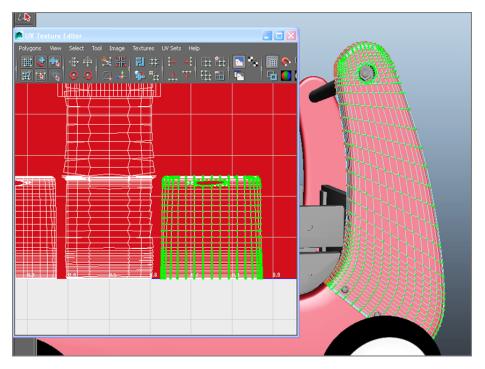
The A panel in the **UV Texture Editor**

This section assumes you have some working knowledge of Adobe Photoshop. You can skip the creation of the maps and use the maps already on the companion web page, which are called out in the text later in the exercise.

The UV Texture window works almost like any other view panel. You may navigate the window and zoom in and out using the familiar Alt+mouse button combinations.

2. RMB+click any part of the wireframe layout in the UV Texture Editor window and select UV to enter the UV selection. Select the entire wireframe mesh at the lower right, as shown in Figure 7.66. Notice that green points are selected—almost as if they were vertices. These are UV points, and they're what define the UV coordinates on that part of the mesh. Look in the perspective view panel; the entire front face of the A panel is selected as well as the green UV points.

Figure 7.66 Select the UVs on this part of the A panel mesh.



Feel free to select parts of the UV layout in the UV Texture Editor to see what corresponding points appear on the mesh in the persp view panel. This will help orient you as to how the UV layout works on this mesh.

3. You need to again lay out just this area of the mesh's UVs. Because you already have the entire front side of the panel selected in UVs, let's convert that selection to poly faces on the model. In the Polygons menu set and in the Maya main menu bar, choose Select → Convert Selection → To Faces. The front faces of the A panel mesh are selected, as shown in Figure 7.67. You can always manually select just the front faces of the mesh, but this conversion method is much faster. The UV Texture Editor shows just those faces now.

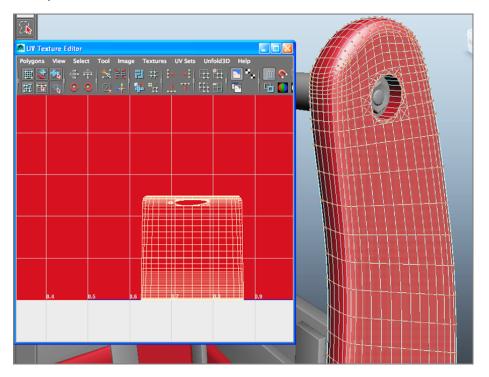


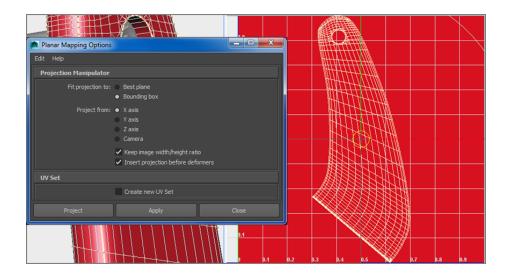
Figure 7.67

Convert the UVs you just selected to a face selection. This method easily isolates the front faces of the A panel for you to lay out their UVs again.

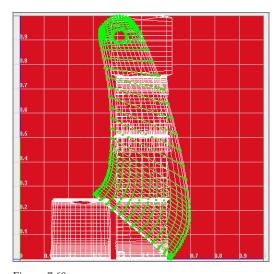
- 4. With those faces selected, make sure you're in the Polygons menu set. Choose Create UVs → Planar Mapping □ from the main menu bar. In the option box, set the Project From option to X Axis, check the Keep Image Width/Height Ratio option, and then click Project. (See Figure 7.68.) The UV Texture Editor shows the front A panel face.
- 5. Now the front face has a much simpler UV layout from which to paint. However, it's centered in the UV Texture Editor and will overlap the other UVs of the same mesh. You should move and size it to fit into its original corner, more or less, to make sure no UVs double up on each other. In the UV Texture Editor, RMB+click the wireframe, and select UV to enter UV selection. Select all the UVs on those faces: all the UVs for the A panel mesh appear in the UV Texture Editor, and you can see the overlap. See Figure 7.69.

Figure 7.68

Create a planar projection for the UV layout.



- 6. Press W for the Move tool and move the selected UVs to the side of the UV Texture Editor. Press R for the Scale tool and scale them down a bit to fit into the corner, as shown in Figure 7.70.
- 7. Earlier in the chapter, you saw how to write out a PSD file with a UV snapshot as one of its layers. You'll use a similar technique to paint the decal for the panel. Click an empty area of the UV Texture Editor to deselect everything. Then, in Object Selection mode (press F8 if you need to exit Component Selection), select the A panel mesh in the persp view panel.





The UVs for the A panel, with the front side's UVs still selected

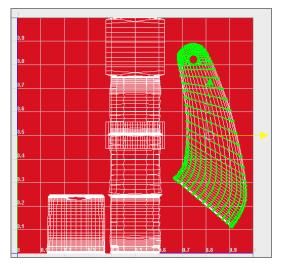


Figure 7.70

Position these UVs to make sure they don't overlap the rest of the A panel mesh's UVs.

Figure 7.71

snapshot

Settings for the UV

8. In the UV Texture Editor window, select Polygons → UV Snapshot to open the option box. Set both Size X and Size Y to 1024. Change the image format to TIFF, click the Browse button at the top next to the File Name field, and navigate to the RedWagon

project's Sourceimages folder on your hard drive. Name the file Apaneluv.tif, and click Save. Leave UV Range set to Normal (0 to 1), and click OK. See Figure 7.71.

WORKING IN PHOTOSHOP

Next, you'll go into Photoshop to paint your map according to the UV layout you just output.

1. In your OS file browser, navigate to the RedWagon project's Sourceimages folder and open the file ApanelUV.tif in Photoshop. Figure 7.72 shows the layout of the UVs that you'll use to create the white stripe for the front of the A panel.

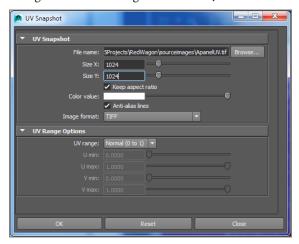
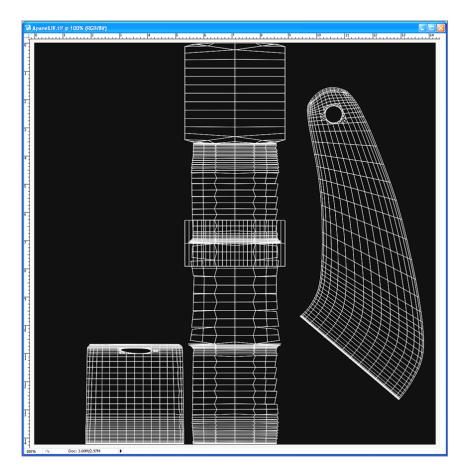


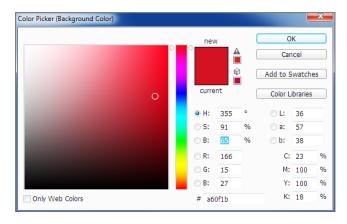
Figure 7.72 Using a UV snapshot makes working with the UV layout for

the A panel easy.



2. In Photoshop, create a new layer on top of the background layer that is the UV layout (white on black, as shown). Using the Bucket tool, fill that new layer with the same red you used on the shader in your scene. To do so, click the foreground color swatch in Photoshop and set H to 355, S to 91 percent, and B to 65 percent, as shown in Figure 7.73. Click OK.

Figure 7.73 In Photoshop's Color Picker, create the same red you used for the wagon.



- 3. Using the Bucket tool, click to fill the entire image with the red you just created. The trouble is that now you can't see the UV layout. Set the Opacity value of the red layer in Photoshop to **50** percent, as shown in Figure 7.74.
- 4. Set Photoshop's foreground color to white. Using the Line and Brush tools set to a width of about 6 pixels, draw a stripe following the UV layout lines, as shown in Figure 7.75. Doing so places that white stripe along the A panel's outer edge because the UV lines you're following correspond to that area of the mesh. The rest will be left red.
- 5. You may have drawn directly on the red layer in Photoshop or created a new layer for the stripe. In either case, set the Opacity value of the red layer back to 100 percent so you can no longer see the UV layout. Your image file should look like the one in

□ □ T □ □

Figure 7.76. Save the image as ApanelStripe.tif in the Sourceimages folder of your RedWagon project. You may keep the layers in the TIFF file, or you may choose to flatten the image or merge the layers. It may be best to keep the stripe and red on separate layers so that you can go back into Photoshop and edit the stripe as needed.

Figure 7.74

Set the opacity for the red layer in Photoshop so you can see the UV layout on the layer below.

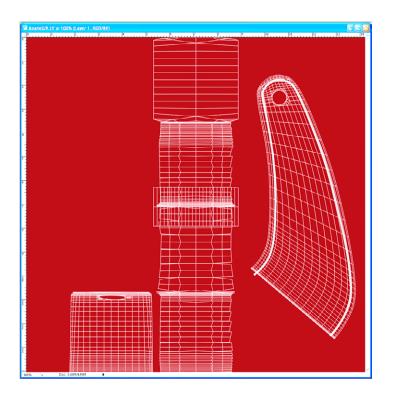


Figure 7.75 Follow the UV lines to draw the white stripe.

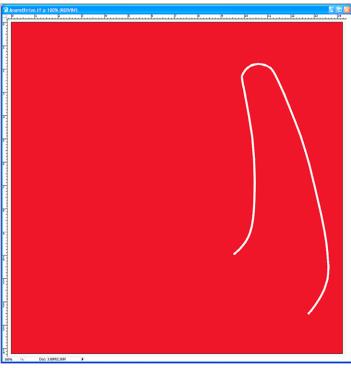
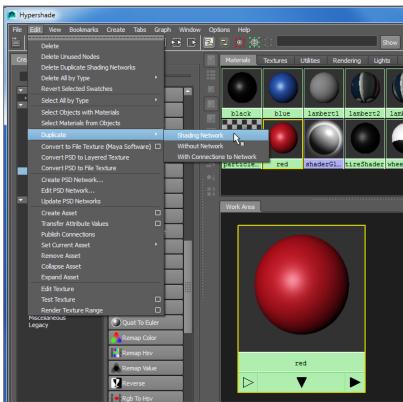


Figure 7.76 The striped image file

CREATING AND ASSIGNING THE SHADER

Now, let's create the shader and get it assigned to the geometry.

- 1. In Maya, open the Hypershade and select the Red shader. Duplicate it by choosing Edit → Duplicate → Shading Network in the Hypershade window, as shown in Figure 7.77.
- 2. Name the new shader (called red1) **ApanelStripe**. Open the Attribute Editor, click the Map icon (▶) next to Color, and choose File. In the Attribute Editor, click the Folder icon next to the Image Name field. Navigate to your Sourceimages folder, and select ApanelStripe.tif, as shown in Figure 7.78.
- 3. In the Hypershade, you may see that the Shader icon has turned somewhat transparent. Maya is automatically mapping the Transparency attribute of the shader as well as the color. Double-click the shader to open its Attribute Editor, RMB+click Transparency, and select Break Connection from the context window. Doing so sets the shader to the same red you used earlier, but now it gives you a stripe along the side A panel.





4. Assign the ApanelStripe shader to the A panel mesh and press 6 for Texture Display mode in the persp view panel. See Figure 7.79. The stripe lines up well.

You may skip the image creation using Photoshop and use the ApanelStripe.tif image file in the Sourceimages folder of the RedWagon project on the companion web page instead.

COPYING UVS

You need to put the stripe on the other side's A panel. Select the other A panel, and assign the ApanelStripe shader to it. You'll notice that no stripe appears. (See Figure 7.80.) This is because the UV layout for this A panel hasn't been set up yet. Don't worry; you don't have to redo everything you did for the first A panel. You can essentially copy the UVs from the first A panel mesh to this one.

1. Select the first A panel (with the stripe) and the second panel (without the stripe). In the Polygons menu set, choose Mesh → Transfer Attributes □. In the option box, set Sample Space to Local, as shown in Figure 7.81, and click Transfer.

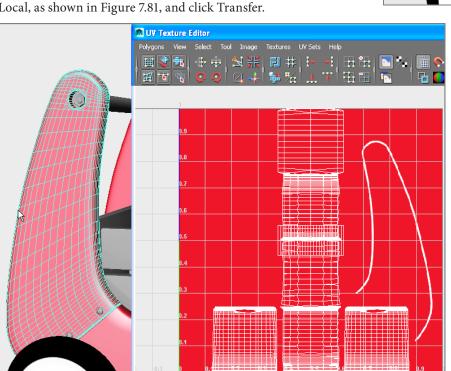


Figure 7.80



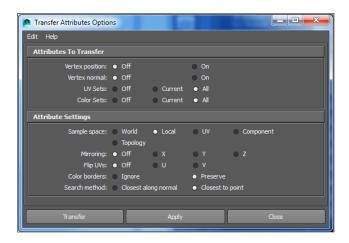


Figure 7.79

The stripe

Figure 7.81

The Transfer Attributes settings



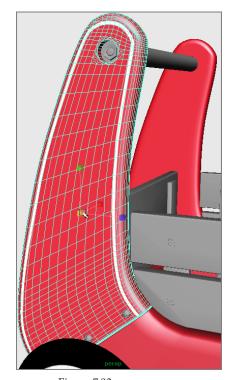


Figure 7.82
The stripe is on the correct side now.

- The stripe appears on the inside of the back A panel, not on the outside as you need. Select that A panel, and choose Modify → Center Pivot.
- 3. In the Channel Box, enter a value of -1.0 for Scale X. The stripe flips to the correct side, as shown in Figure 7.82.
- 4. With that A panel still selected, choose Modify → Freeze Transformations.

You can load the file RedWagonTexture_v02.ma from the Scenes folder of the RedWagon project to check your work or skip to this point.

The file texture you'll use for the panels was painted in Photoshop to place the stripes and logo properly on the wagon using their UV layouts. Study the image file, and see how it fits on the mesh of the wagon. Try adjusting the image file with your own artwork to see how your image map affects the placement on the mesh.

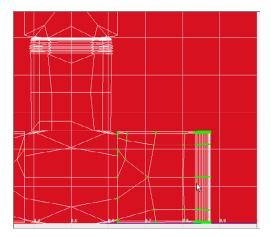
Working with the B Panels

With the A panels done, you'll move on to the B panels (Figure 7.36 earlier in the chapter) and use much the same methodology you did with the A panels. To begin, follow these steps:

1. Select one of the B panels, shown in Figure 7.83, and open the UV Texture Editor window.

Figure 7.83 Starting on the **B** panels

- 2. As you did with the front face of the A panel, select the UVs on the lower-right side of the layout in the UV Texture Editor, as shown in Figure 7.84, to isolate the front face of the B panel.
- 3. Choose Select → Convert Selection → To Faces. You've isolated the front face of the B panel.
- 4. Choose Create UVs \rightarrow Planar Mapping \square . In the option box, set the Project From option to X Axis, and make sure the Keep Image Width/Height Ratio option is checked, just as before. Your B panel shows up nicely laid out in the UV Texture Editor. See Figure 7.85.





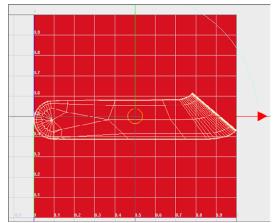


Figure 7.85 The planar projection creates a nice UV layout for the

front faces of the B panel.

Figure 7.86

Put the front face UVs on the side.

- 5. Convert the selection to UVs and use Move (W), Scale (R), and Rotate (E) to position the UV layout for that front face, as shown in Figure 7.86.
- 6. Press F8, and select the B panel mesh. In the UV Texture Editor, save a UV snapshot called **BpanelUV.tif** to the Sourceimages folder of the RedWagon project.
- 7. Open the Bpaneluv.tif image in Photoshop and follow the same steps as you did for the A panel to lay down a red layer and paint a stripe along the layout, as shown in Figure 7.87. It's best to save the stripe on its own layer in Photoshop because you'll probably need to edit and reposition the stripe to make sure it lines up with the A panel stripe after you assign the shader.
- 8. Create your own logo to place in the middle of panel B and place it in the Photoshop image file, as shown in Figure 7.88. Save the image file as BpanelStripe.tif into the Sourceimages folder.

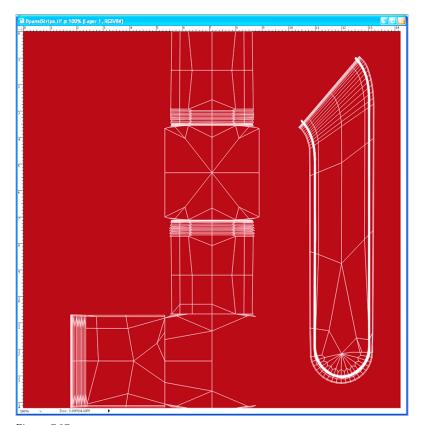


Figure 7.87

Create the B panel's stripe in Photoshop using the UV snapshot.



Create the logo in Photoshop.

- 9. Duplicate another Red shader and, as you did previously, assign the BpanelStripe.tif as its color map. If necessary, disconnect the transparency from the shader as you did with the A panel's shader. Name the shader **BpanelStripe**.
- 10. Assign the BpanelStripe shader to the B panel, as shown in Figure 7.89.



Figure 7.89 The B panel has its decals.

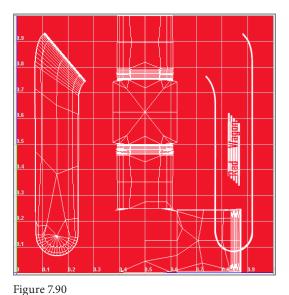
You can skip the image creation in Photoshop and use the BpanelStripe.tif image file in the Sourceimages folder of the RedWagon project on the companion web page.

CREATING THE OTHER B PANEL TEXTURE

Finally, you need to create the shader for the other side's B panel. Assign the BpanelStripe shader to the other B panel. Nothing happens, because the UVs for the second B panel aren't set up yet.

However, because there is a logo with text, setting up its UVs won't be as simple as copying the UVs from the first B panel and then mirroring the mesh, as you did with the A panel with a Scale X value of -1.0. Doing so will make the logo and text read backward. First you'll copy and flip the UVs to the other B panel.

1. Select the first B panel with the correct texture; then select the other side's B panel and choose Mesh → Transfer Attributes □. Make sure Sample Space is still set to Local, and set Flip UVs to U. Click Transfer to copy the UVs, flipping them over, as you can see in Figure 7.90.



Copying and flipping the UVs to the other B panel

- 2. You have to go back to Photoshop and create a second BpanelStripe.tif image file with a mirrored logo. In Photoshop, create a marquee around the logo and mirror or flip the canvas horizontally. See Figure 7.91.
- 3. Select the logo portion of the image and flip that vertically, as shown in Figure 7.92. Save the image as BpanelStripe_2.tif.
- 4. Duplicate the BpanelStripe shader in the Hypershade by selecting the shader and choosing Edit → Duplicate → Shading Network. The copy is called BpanelStripe1.
- 5. Select the newly copied BpanelStripe1 shader and graph its input and output connections () in the Hypershade. Select its file node and open the Attribute Editor. Click the Folder icon to select a new image file and then select the BpanelStripe_2.tif you just created in the Sourceimages folder. See Figure 7.93.

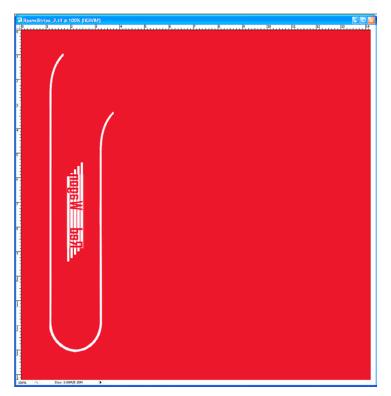


Figure 7.91

Flip the original image horizontally to fit the new UV layout of the second B panel.

file1 place2dTexture7

▼ File Attributes

- 6. The stripe and logo display on the wrong side of the B panel. Select the mesh and center its pivot.
- 7. Set the Scale X attribute for the B panel to -1.0 to mirror it. The stripe and logo decals now show up on the correct side of the panel. Select the mesh and freeze its transforms. Figure 7.94 shows the wagon so far.

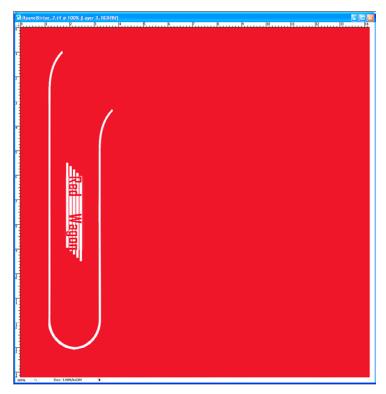


Figure 7.92 Flip the logo vertically and save the image as its own file.



Lock File Node Name To Image Nam Figure 7.93

Assign the new image file to the new BpanelStripe1 shader.

Figure 7.94

The wagon has decals on both sides.

Texturing the Floor

Right now, the floor of the wagon is red, like the rest of its body. However, the real wagon has a blue floor, not red. If you select the mesh for the wagon's floor (named wagonFloor) and assign the Blue shader you created, the whole body of the wagon turns blue, and that isn't what you want. You need only the inside and bottom of the floor to be blue, not the front and back sides of the wagon's body.

You'll make a face assignment instead of dealing with UVs and image files. RMB+click the wagon floor mesh and select Face from the marking menu. Select the two faces for the floor, as shown in Figure 7.95.

Figure 7.95 Select the floor faces.



With the faces selected, assign the Blue shader from the Hypershade window, and you're finished! You have a blue floor. All that remains now are the screws, bolts, handle, and wood railings.

You can load the file RedWagonTexture_v03.ma from the Scenes folder of the RedWagon project to check your work or skip to this point.

Shading the Wood Railings

You'll use procedural shading to use the Wood texture available in Maya to create the wood railings. Begin here:

1. In the Hypershade, create a new Phong material.

- 2. Click the Color Map icon () and choose the Wood texture from the 3D Textures heading in the Create pane in the Hypershade.
- 3. In the Attribute Editor for the Wood texture, set the Filler Color and Vein Color attributes according to Table 7.3.

ATTRIBUTE	H VALUE	SVALUE	V VALUE
Filler Color	43	0.25	1.0
Vein Color	10.7	0.315	0.9

Table 7.3 Filler Color and Vein Color attributes

- 4. Set Vein Spread to **0.5**, Layer Size to **0.5**, Randomness to **1.0**, Age to 10.0, and Grain Contrast to 0.33. In the Noise Attributes heading, set Amplitude X to 0.2 and Amplitude Y to 0.1, as shown in Figure 7.96. Name the shader **Wood**.
- 5. Select all the wood railings and posts and assign the Wood shader to them. Render a frame and compare it to Figure 7.97. Notice the green cube place3dTexture node that is now in your scene. See Figure 7.98.
- 6. The side wood railings look fine; however, the wavy pattern on the front and back wood railings looks a bit odd. In the Hypershade, duplicate the shading network for the Wood shader and call the new shader woodFront.
- 7. Assign that shader to the front and back railings and posts. Graph the network on the woodFront shader in the Hypershade window.
- 8. In the Hypershade, select the place3dTexture2 node (Figure 7.99) and the green cube in the view panels.



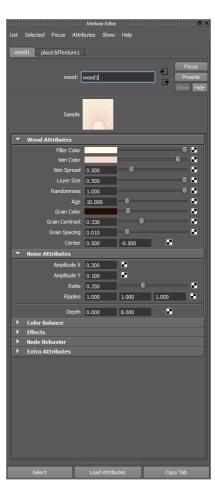


Figure 7.96 Setting the Wood texture

Figure 7.97 Assign the Wood shader.

Figure 7.98

The place3dTexture node for the wood texture

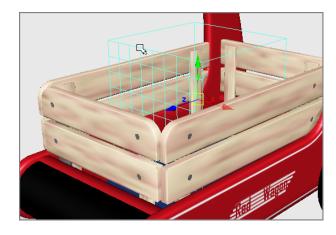
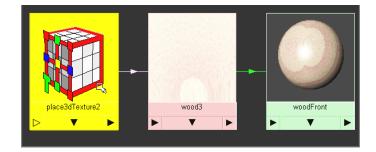


Figure 7.99

Select the placement node for the second wood texture.



9. Rotate that placement node in the persp view panel 90 degrees to the right or left. Render a frame, and compare it to Figure 7.100. The wood should no longer have that awkward wavy pattern.

Figure 7.100

The wood on the front and back railings looks better.





Figure 7.101 A nice bump for the bullnose

The wood railings are finished. Now, for some extra challenge, you can use pictures of real wood to map onto the railings for a more detailed look. The procedural Wood texture can give you only so much realism. If you create your own wood maps, use your experience with the side panels to create UV layouts for the railings so you can paint realistic wood textures using Photoshop. You'll use custom photos and texture image maps next to simulate the rich wood in the decorative box later in the chapter.

Finishing the Wagon

Now that the railings are done and you have test renders, there are only two parts left to texture: the bullnose front of the wagon and the metal handle and screws. From here, take your time and create a bump map based on a fractal, as you did for the tires, and apply it to the bullnose's black shader. Figure 7.101 shows a nice subtle bump map on the bullnose.

And last, you'll need a metal shader for the screws, bolts, and handlebar for the wagon. Use a Phong shader with a blue-gray color and a low diffuse value and assign it to all the metal parts of the wagon, as shown in Figure 7.102. You can then add an environment map to the reflection color like you did for the table lamp stem earlier in this chapter to give the metal a reflective look.

Because metal is a tricky material to render and a lot of a metal's look is derived from reflections, you'll finish setting the Metal shader's attributes in Chapter 11, when you render the wagon as well as the table lamp. You'll enable raytracing to get realistic reflections and gauge how to best set up the Metal shader for a great look.

Figure 7.103 shows the wagon with all its parts assigned to shaders. Figure 7.104 shows a quick render of the wagon as it is now. Figure 7.102

> Select all the metal screws, the bolts, and the handlebar and assign the Metal shader to them.







Figure 7.103

The wagon in the Perspective panel

Figure 7.104 A current render of the wagon

You can load the file RedWagonTexture_v04.ma from the Scenes folder of the RedWagon project to check your work or skip to this point.

Photo-Real Mapping: The Decorative Box

With all the references you can find to any given object on the Internet, why not use real photos to create the textures for a model? That's exactly what you'll do here, with the decorative box you modeled in Chapter 3, using pictures of the real box.

You'll take this texturing exercise one important step further in Chapter 11 and experience how you can add detail to an object through displacement mapping, after you assign the colors in this chapter. This will allow you to add finer detail to a model without modeling those details.

Setting Up UVs (Blech!)

The UVs on the decorative box aren't too badly laid out by default, as you can see in Figure 7.105. The only parts of the box that are missing in the UV layout are the feet. That is a common issue when extruding polygons: Their UVs are rarely laid out automatically as you extrude them. Frequently, they're bunched up together in a flat layout that is difficult, if not impossible, to see in the UV Texture Editor.

Open the file boxModel_v07.mb in the Scenes folder of the Decorative_Box project on the companion web page, or continue with your own model.

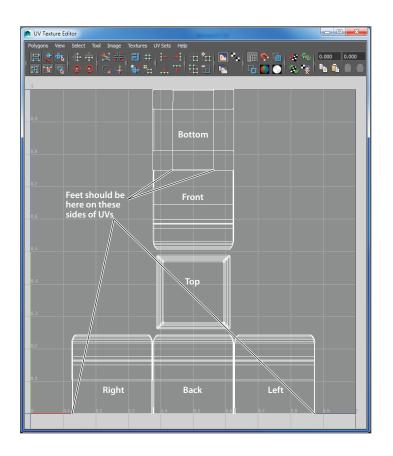


Figure 7.105

The feet UVs are missing from the box model.

First you have to make room for the feet UVs.

- 1. Select all the UVs for the entire box in the UV Texture Editor. Press R for the Scale tool and scale everything down uniformly to gain some space in the normalized UV space, shown in Figure 7.106.
- 2. Now for the tedious part. You have to create new UVs for the four feet and then move them to where they should be in the full box's UV layout. Go into Component mode and select the poly faces for the four feet, as shown in Figure 7.107.
- 3. With the faces selected, go to the main Maya window. In the Polygons menu set, select Create UVs → Automatic Mapping. The feet now have UVs that you can see in Figure 7.108. However, they're all over the place. You have to individually select and move each face of each foot to its appropriate place on the box's overall UV layout.

Figure 7.106

Scale all the UVs
down a bit.

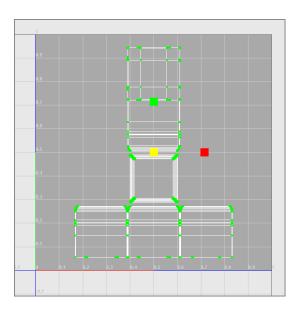
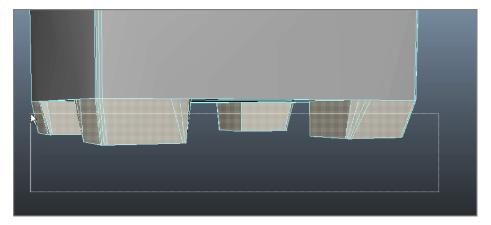


Figure 7.107

Select the faces for the feet.



4. Select all the feet UVs in the UV Texture Editor and scale them all down uniformly together. You want to scale them all so that one of the squares fits into the corner of the top shape, as shown in Figure 7.109 (left). Then move them all together off to the side, as shown in Figure 7.109 (right). You'll position and scale them to fit properly soon.

Laying out UVs can be a time-consuming affair, as you've seen with the wagon. Although it's recommended that you follow along with this exercise to lay out UVs for the box (because doing so will give you more practice and experience with UVs), you can skip straight to color-mapping the box in the next section by downloading the file boxTexture01.mb in the Scenes folder of the Decorative_Box project on the companion web page.

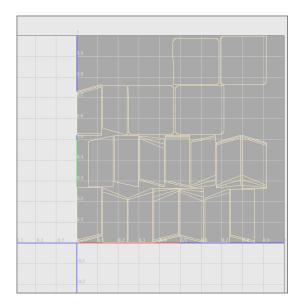


Figure 7.108 The automatic UV creation puts the

UVs everywhere.

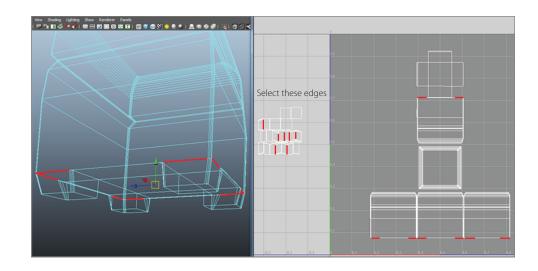
Figure 7.109

Scale the feet UVs to fit the size of the rest of the box's UVs (left). Then move the feet UVs to the side of the UV editor (right).

5. Now comes the task of figuring out which UV fits where. In the UV Texture Editor window, select the edges shown in Figure 7.110 by Shift+single-clicking them. Do not use a marquee selection to make sure you select only the single edges you need. Notice that as you select each edge, they become selected in the model as well as the feet UVs off on the side of the UV Texture Editor window. These are the edges where the outside sides of the feet meet the box. If you find it easier, you can select those edges from the model itself instead of the UV Texture Editor.

Figure 7.110

Select these individual edges either in the model view or in the UV Texture Editor.



6. With those edges selected, in the UV Texture Editor choose the Move And Sew The Selected Edges icon (Figure 7.111). This will snap the UVs for the outside sides of the feet to their respective sides on the box. If you did not scale the feet UVs in step 4 well enough, you will notice that the feet will stick out from the sides of the box's UVs, as shown in Figure 7.112 (left). If that is the case, it's easier to return to step 4 (press Z for

Figure 7.111

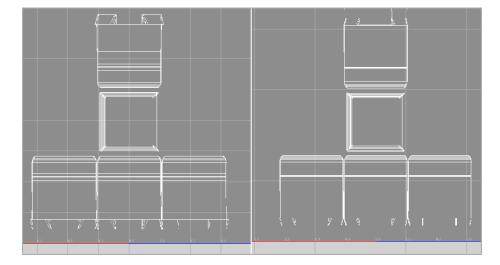
The Move And Sew The Selected Edges icon in the UV Texture Editor

Figure 7.112

Sew the feet edges to the box. On the left, the feet UVs are scaled too big in step 4. On the right, the feet UVs are scaled properly.



undo until you reach step 4 again) and scale those feet UVs more accurately. This can have a little bit of a back and forth. As long as you are within reasonable size, you will be fine. Figure 7.112 (right) shows a better matching scale of the feet after the UVs are sewn together.



7. RMB+click in the UV Texture Editor and select UV from the marking menu. Next, select just one corner UV from the bottom shape of the box, as shown in Figure 7.113 (left). In the UV Texture Editor, choose Select → Select Shell to select the entire bottom UV shape and then press W to move the bottom UVs up in the UV Texture Editor a little bit, as shown in Figure 7.113 (right).

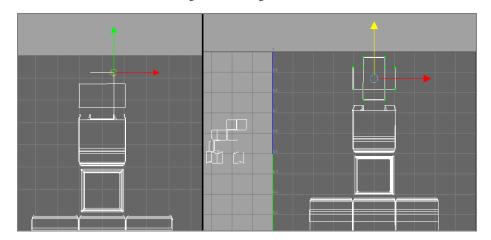


Figure 7.113

Select one UV (left); then select the shell and move the bottom UVs up in the **UV Texture Editor** (right).

8. Figure 7.113 (right) also shows the remaining feet UVs off on the left side of the UV Texture Editor window. The four square shapes are the bottoms of the feet. Select just one corner UV of the square shown in Figure 7.114 (left); then select the entire shell (Select → Select Shell) and move it to fit to the bottom UVs of the box as shown in Figure 7.114 (right).

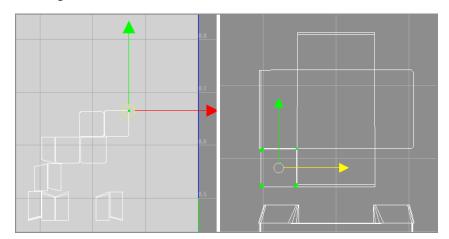


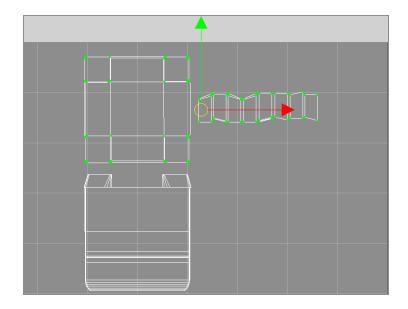
Figure 7.114

Select one UV in the corner (left): then select the shell and move it to fit the box bottom UV shape.

9. Repeat step 8 to place all the feet bottom shapes around the UVs of the bottom of the box, as shown in Figure 7.115.

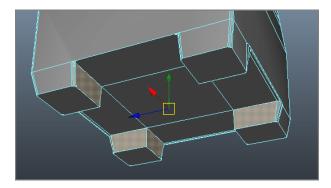
Figure 7.115

Place the feet bottoms to the box bottom and then line up the remaining feet UVs to the side.



10. Individually select the shells of the remaining feet UVs on the side of the UV Texture Editor and line them up to the side of the box bottom as shown in Figure 7.115. Those remaining feet UVs are the inside faces of the feet and share the same generic wood texture as the bottom of the box, so aligning and sewing them together like you did for the outside faces of the feet in steps 5 and 6 is not necessary (Figure 7.116).

Figure 7.116 These faces will have a generic wood texture like the bottom of the box and do not need to be carefully lined up.



When you've finished, your UV Texture Editor should resemble the one shown in Figure 7.117. Because the box's decorations are seamless from the top of the box down to the four sides, let's lay out the UVs to make painting and editing in Photoshop easier.

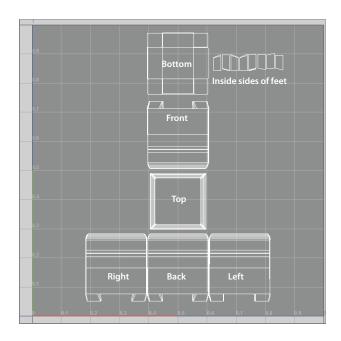


Figure 7.117 Finally, you're finished with the feet UVs.

In the UV Texture Editor, select the box top's edges, as shown in Figure 7.118 (left). Then click the Move And Sew The Selected Edges icon (shown earlier in Figure 7.111) in the UV Texture Editor, and your UV layout should match Figure 7.118 (right). The top and sides of the box are all one UV shell now.

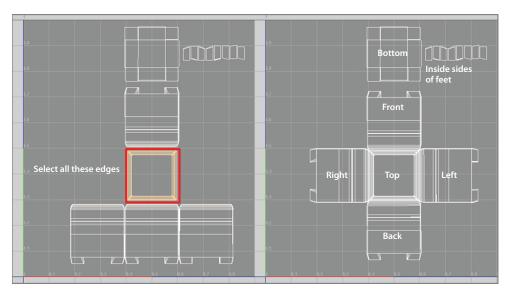


Figure 7.118 Select the box top edges (left) and sew them together (right).

UV layout can be an exacting chore, but when it's completed, you're free to lay out your textures. You can check your work against the file boxTexture01.mb in the Scenes folder of the Decorative Box project on the companion web page. You can also take a much needed breather. I sure hope you've been saving your work!

Color Mapping the Box

Now that you have a good UV layout, you can output a UV snapshot and get to work editing your photos of the box to make the color maps. Start with the following steps:

1. Select the box and open the UV Texture Editor window. From the UV Texture Editor menu, select Polygons → UV Snapshot. In the UV Snapshot window, set Size X and Y to 2048. Change Image Format to TIFF.

Click the Browse button and select a location for your UV snapshot image. Generally, the project's Sourceimages folder is the best place for it. Make sure you don't write over the UV snapshot already created for you. Type in a name for your UV snapshot and click OK to create the image. Figure 7.119 shows the option box, and Figure 7.120 shows the UV snapshot image.

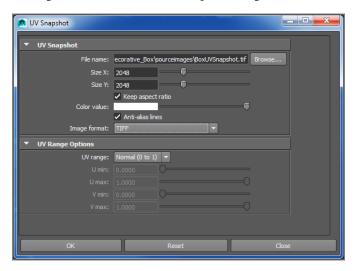
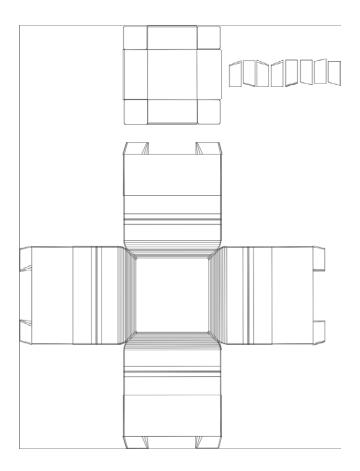


Figure 7.119 Setting the UV **Snapshot options**

> 2. Open the UV snapshot image in Photoshop or your favorite image editor and set it as its own layer. Rename the layer to **UV Snapshot**. I've done the heavy lifting for you and have prepared five photos of the decorative box that you can use to map the model. Figure 7.121 shows the photos of the box. This image file is included as lineup.jpg in the Sourceimages folder in the Decorative_Box project on the companion web page.



Front

Back

Figure 7.120

The UV snapshot for the decorative box, shown as black lines on white. You may see white lines on black in Photoshop.



Figure 7.121 Photos of the box

3. As you've probably guessed, you need to copy and paste the photos to their respective views over the UV Snapshot layer. Open the lineup.jpg file in Photoshop alongside the UV snapshot. Marquee+select a box around the top image (the one at the left in Figure 7.121), and copy it (Ctrl+C or Edit → Copy in Photoshop).

Left

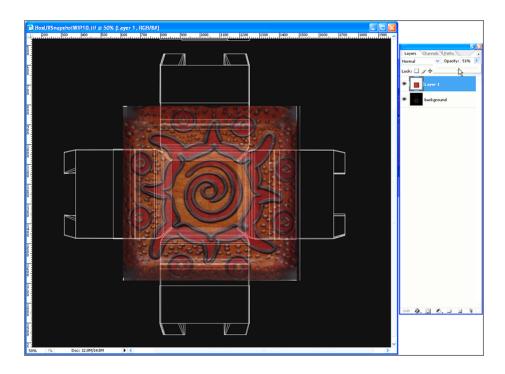
Top

Right

4. Go to the UV snapshot image in Photoshop and paste the image on top. Rename the new layer to Box Top and set the layer's Opacity to 50% so you can still see the UV layout, as shown in Figure 7.122.

Figure 7.122

Paste the top image onto the UV snap-shot image.



5. Use the Scale function in Photoshop (Edit → Transform → Scale) to move and scale the top image to fit over the top of the UV layout, as shown in Figure 7.123. Make sure you scale the box-top image uniformly to keep it from distorting. You can do this by holding the Shift key as you scale the image up or down.

These photo images of the box have been retouched and painted to create an overlap. This means that parts of the sides of the box show in the top image. As you can see in Figure 7.123, the top image extends slightly all around the four sides. This allows the different parts of the texture map (top and four sides) to overlap and blend with each other better when put on the model.

Save your work as **boxColorMap01.psd** in the project's Sourceimages folder. Saving as a Photoshop file will preserve the layers for easier editing.

- 6. Marquee+select the right-side image of the box (immediately to the right of the top image in lineup.jpg) and copy it. Paste it into boxColorMap01.psd in Photoshop. Do your best to align the right-side image with the top image, using the features of the box to line them up, as you can see in Figure 7.124. You can fix this later by adjusting both the map and the UVs on the box for a tighter fit. For now, be fairly accurate and leave the finesse for later. Save the file.
- 7. Use the same procedures in Photoshop to line up the other sides of the box, as shown in Figure 7.125. Set the box-top image to be the topmost layer, make sure all the layers are at 100% opacity, and then turn off the UV Snapshot layer so it's not visible.

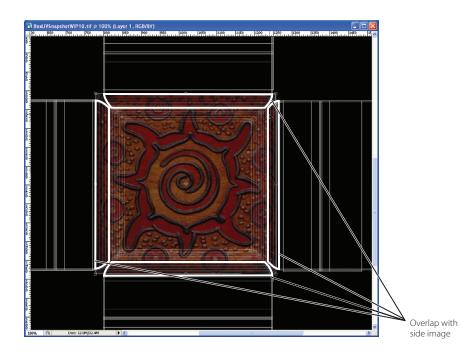


Figure 7.123

Position and scale the top image in Photoshop to line up with the UVs of the top of the model. Notice the overlap of the sides and the top.

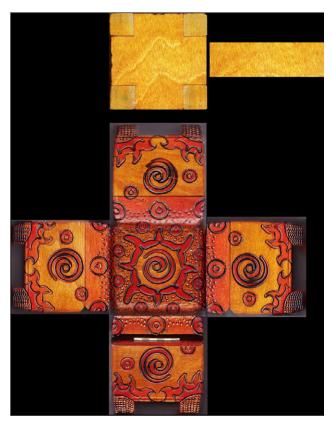


Figure 7.124

Align the right-side image with the right-side UVs.

Figure 7.125

Copy, paste, and line up the box sides and the back to their respective UV areas.



- 8. And finally, open the file boxBottomAnd-Feet.jpg from the Sourceimages folder of the project. Select the entire image and copy and paste into your boxColorMap01. psd file. Position and place the image to fit the bottom UVs of the box in your map in Photoshop. Just get close; you will fine-tune the placement of image to UV later.
- 9. Save the final Photoshop file. Then, resave the file as a JPEG called boxColorMap01.jpg. This is the file you'll map. (See Figure 7.126.)

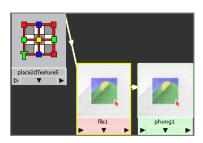
Mapping the Box

Let's map this color image to the box and see how it fits. Based on rendering the box, you can make adjustments to the UVs and the image map to get everything to line up. This, of course, requires more Photoshop and/or image-editing experience, which could be a series of books of its own. If you don't have enough image-editing experience, have no fear: The images have been created for you so you can get the experience of mapping them and learn about the underlying workflow that this sort of texturing requires. Follow these steps:

- In Maya, open the Hypershade window and create a new Phong shader. Open the Attribute Editor, click the Map icon (next to the Color attribute, and select File.
- 2. Double-click the file1 node to open the Attribute Editor. Click the Folder icon next to the Image Name attribute, navigate to the Sourceimages folder for the project, and select the boxColorMapO1.jpg file (not the PSD file). The icon in the Hypershade doesn't show the image because it's a large file. (See Figure 7.127.)

Figure 7.126

3. Double-click the file1 node and in the Attribute Editor rename it to **boxColorMap**. RMB+click the newly named boxColorMap node in the Hypershade and choose Refresh Swatch from the marking menu. The boxColorMap image is displayed as the boxColorMap icon. (See Figure 7.128.)



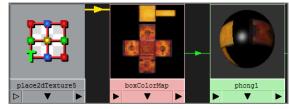


Figure 7.127

The color map's file node

Figure 7.128

The icon is refreshed.

- 4. Select the box and assign the Phong shader to it. Rename the Phong to **boxShader**. In the persp view panel, press 6 for Texture view. The color map is fairly well aligned on the model. Not bad! (See Figure 7.129.)
- 5. Render a frame to see how the box looks. Notice that there are small alignment issues at the edges where the sides meet and where the top meets the sides. Save your Maya scene.

This gives you a pretty good place to work from. You need to adjust the color map image to be more seamless. The scene file boxTexture02.mb in the Scenes folder of the Decorative_Box project on the companion web page will catch you up to this point.

Figure 7.129

The color map fits pretty well already, but there are a few alignment issues at the edges.

Photoshop Work

This is where image-editing experience is valuable. Here, it's all about working in Photoshop to further line up the sides to the top and the sides to each other to minimize alignment issues and yield a seamless texture map. Although I won't get into the minutia of photo editing here, I'll show the progression of the images and the general workflow used in Photoshop to make the color map's different sides and top line up or merge better. The images have already been created and are on the companion web page in the Sourceimages folder for this project.

For example, using masking in Photoshop, spend some time feathering the intersection of the box's sides in boxColorMap01 .psd so there is no hard line between the different sides and



the top. Figure 7.130 shows a smoother transition between the different parts. This image has been created for you; it's boxColorMap02.jpg in the Sourceimages folder. Make sure you don't overwrite that file if you're painting your own.

Figure 7.130 Use masks to feather the transitions between the different parts of the box.

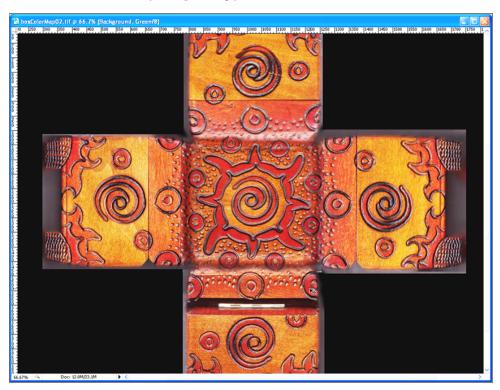


Figure 7.131 There are blank areas on the box as well as a little distortion.



In Maya, double-click the boxColorMap node in the Hypershade and in the Attribute Editor replace the original boxColorMap01.jpg file with the boxColorMap02.jpg file from the Sourceimages folder (or your own retouched image file). Render and compare the difference. The top and front should merge a little better. In the persp view panel, orbit around the box in Texture View mode (press 6) to identify any other alignment issues. In some cases, as you can see in Figure 7.131, gray or black is mapped onto the box on its right side, and there is a warped area. Also, the crease where the lid meets the box is lower than you've modeled.

The blank areas on the box are outside the bounds of the image in the Photoshop image and can be fixed by adjusting the UVs in Maya. The same goes for the distorted areas on the side of the box—you just need to adjust the UVs.

1. Select the box and open the UV Texture Editor window. Figure 7.132 shows some primary areas for you to work on.

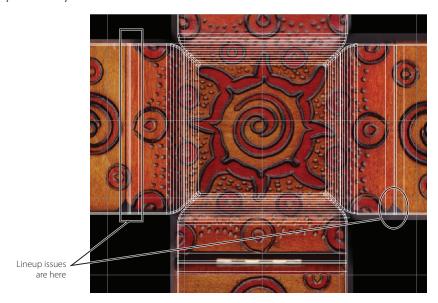


Figure 7.132

Here are the main problems to fix.

- 2. In the UV Texture Editor, select the UVs (RMB+click and choose UV) shown in Figure 7.133 on the left. Press W for the Move tool and realign the UVs to the seam where the lid meets the box, as shown in Figure 7.133 on the right. As you make the changes in the UV Texture Editor, you should immediately notice them in the Perspective window (as long as you're in Texture View mode).
- 3. Look at the image on top in Figure 7.134. Move the appropriate UVs to align the edge of the UV layout to the image for the right side of the box, as shown on the bottom of Figure 7.134.
- 4. The distortion is getting better, and the texture fits nicer. But notice in the image on the left of Figure 7.135 that the right side of the box and the back of the box don't line up well. There are some other instances of the textures not lining up well all around the box. Using the Texture View mode (press 6) in the Perspective window and the UV Texture Editor, go around the box in its entirety and adjust the UVs, point by point as needed, so that they all line up to the image in the UV Texture Editor. Make sure that

the sides line up at the edges of the box as well as you can. The image on the right of Figure 7.135 shows correctly aligned UVs for the right side/back side of the box. Don't forget to line up the bottom of the box and feet, too. That part should be much easier.

This part of tweaking the UVs takes time and patience. The key is to keep looking back and forth between the UV Texture Editor and the persp panel to see how the textures are lining up as you move the UVs. Figure 7.136 shows the UV Texture Editor and a persp view of the box with UVs lined up and reasonably ready to go. You can compare your work to the scene file boxTexture03.mb in the Scenes folder of the Decorative_Box project on the companion web page. Render a few different views to take in all the hard work. In Chapter 10, you'll light the box and prepare it for rendering, and in Chapter 11, you'll use displacement maps created from these photos to detail the indentations and carvings that are in the actual box. You've had enough excitement for one chapter.

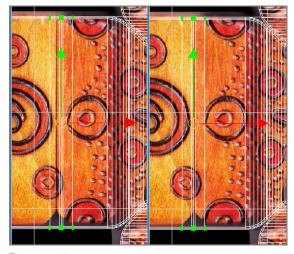


Figure 7.133

Move the UVs.

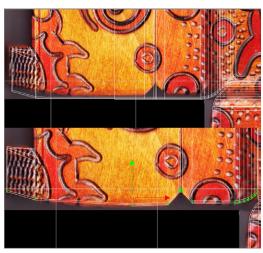


Figure 7.134

Line up the UVs to the image for the right side of the box.



Figure 7.135

Line up the UVs for the right-side/back-side edge of the box.

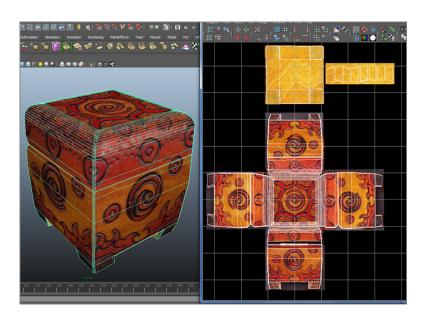


Figure 7.136
The UVs laid out for the decorative box

Toon Shading

Not everything needs to be textured to look real. Maya has a neat shading system that gives you a cartoon look using clever shaders and a powerful feature called Paint Effects, which you will look at a bit more in Chapter 12. The toon-shading system makes your scenes and animations render more like traditional cartoons with flat colors and outlines. Next, you'll take a quick look at how to apply toon shading to the wagon you textured earlier in this chapter to make it render more like a cartoon.

Try This Set your project to the RedWagon project, and open the RedWagonModel.ma scene file from the Scenes folder.

- 1. You'll see the wagon in a 3/4 view in the persp view panel. Select all the parts of the body of the wagon without the railings or wheels, as shown in Figure 7.137.
- 2. Switch to the Rendering menu set and select Toon → Assign Fill Shader → Shaded Brightness Two Tone. The Attribute Editor opens, focused on a Ramp shader, as shown in Figure 7.138.
- 3. Under the Color heading, set the color of the gray part of the ramp to a dark red and set the white part of the ramp to a bright red. Your wagon should turn red in the persp view panel if you're in Shaded or Texture view. Render a frame, and you should see the wagon in only two tones of red but with gray railings and wheels (shown in grayscale in Figure 7.139).
- 4. Select the rail objects, and select Toon → Assign Fill Shader → Shaded Brightness Two Tone to create another Toon shader. Set the colors to a dark tan and a bright tan color in the Color ramp.

Figure 7.137

Select the main body of the wagon, without the wheels or the rails.

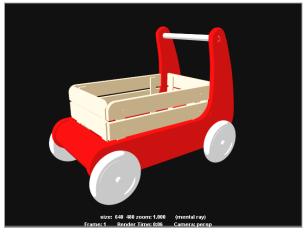


A Ramp shader is added to the scene and applied to the wagon's body.

- 5. Select the handlebar and all four wheels and create another two-tone fill shader with a gray and white Color ramp (which is the default).
- 6. The frame, railings, and wheels now have a Toon shader as well. Notice the wheels in Figure 7.140 and how cool they look when toon-shaded. Of course, adjust any of the colors to your liking.
- 7. Now for the toon outlines. Select all the parts of the wagon, and select Toon → Assign Outline → Add New Toon Outline. A black outline appears around the outside of the wagon's parts, and a new node called pfxToon1 appears in the Outliner or Hypergraph. The outlining is accomplished with Paint Effects.



The wagon now has a toon-shaded body. The front side of the wagon is a darker red than the front because of the lighting in the scene.



 $\frac{Figure~7.140}{\mbox{The wagon has Toon shaders for the fill color applied.}}$

- 8. Before you render a frame, set the background to white to make the black toon outlines pop. To do so, select the persp camera and open the Attribute Editor. Under the Environment heading, set Background Color to white.
- 9. Open the Render Settings window and make sure that the renderer is set to Maya Software. Render a frame and compare your work to Figure 7.141. The outlines are too thick!

The mental ray renderer doesn't render Paint Effects by default. You have to render with Maya Software rendering to be able to see Paint Effects strokes. To see Paint Effects with mental ray, you must convert the paint effects to polygons, a procedure only briefly mentioned in this chapter.

10. Select the pfxToon1 node in the Outliner and open the Attribute Editor window. Click the pfxToonShape1 tab to open the attributes for the outlines. Set the Line Width attribute to **0.03** and render a frame. Compare your work to Figure 7.142.

Adjust the toon outline thickness the way you like and have some fun playing with the toon outline's attributes to see how they affect the toon rendering of the wagon. As you can see, there are a lot of attributes for the toon outlines. Paint Effects is quite a complex system, but after you experiment with the toon outlines, it'll make much more sense. You will return to Paint Effects in Chapter 12 for another quick look at that wonderful system.

Feel free to adjust the colors and the ramps of the fill shader to suit your tastes, and you can easily try the other Fill shader types, such as a three-tone shader to get a bit more detail in the coloring of the wagon. This is just a quick primer to get you into toon shading. The rest, as always, is up to you. With some playing and experimenting, you'll be rendering some pretty nifty cartoon scenes in no time.

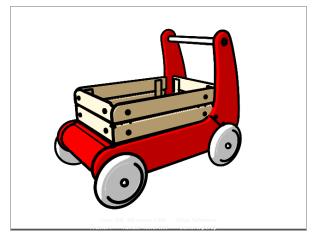


Figure 7.141
The black outlines are applied, but they're too thick.

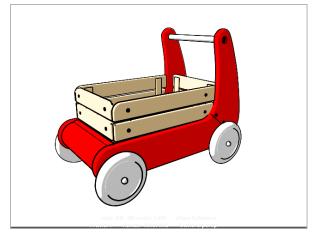


Figure 7.142

The cartoon wagon

For Further Study

For a challenge and more experience, create new image maps for the wagon and try your own decal designs with more traditional (non-toon) renders. You can even change the textures for the decorative box and make your own design. As previously suggested, you can try to create more realistic wood maps for the wagon's railings. In Chapter 10, you'll begin to see how shading and rendering go hand in hand; you'll adjust many of the shader attributes you created in this chapter to render the toy plane and decorative box in Chapter 11.

Summary

In this chapter, you learned about the types of shaders and how they work. Each shader has a set of attributes that give material definition, and each attribute has a different effect on how a model looks.

To gain practice, you textured the toy plane scene using a couple of shaders and texture nodes.

Next, you learned about the methods you can use to project textures onto a surface, and you learned about the Maya texture nodes, including PSD networks and the basics of UVs, and how to use them to place images onto your wagon and decorative box models in detailed exercises exposing you to manipulating UVs and using Photoshop to create maps. And finally, you took a quick look at toon shading and created a cartoonish render of the wagon.

Texturing a scene is never an isolated process. Making textures work involves render settings, lighting, and even geometry manipulation and creation. Your work in this chapter will be expanded in Chapters 10 and 11 with discussions of lighting and rendering.

The single best weapon in your texturing arsenal, and indeed in all aspects of CG art, is your eye; specifically, your observations of the world around you and how they relate to the world you're creating in CG.



Introduction to Animation

The best way to learn about animation is to start animating, so you'll begin this chapter with the classic exercise of bouncing a ball. You'll then take a closer look at the animation tools the Autodesk® Maya® software provides and how they work for your scene. You'll do that by throwing an axe. Finally, you'll tackle animating a more complex system of parts when you bring a locomotive to life.

Learning Outcomes: In this chapter, you will be able to

- Set keyframes to establish the movement scheme for an object
- Import objects into a scene
- Create the feeling of weight and mass for an animated object using scale animation
- Read animation curves in the Graph Editor
- Differentiate among different animation principles such as squash, stretch, anticipation, and follow-through
- Set up a hierarchy for better animation control
- Transfer animation between objects
- Create text
- Create motion trails and animate objects along a path
- Set up models for animation
- Use selection handles to speed up workflow
- Animate objects in time with each other

Keyframe Animation: Bouncing a Ball

No matter where you study animation, you'll always find the classic animation exercise of creating a bouncing ball. Although it's a straightforward exercise and you've probably seen it a hundred times on the Web and in other books, the bouncing ball is a perfect exercise with which to begin animating. You can imbue the ball with so much character that the possibilities are almost endless, so try to run this exercise as many times as you can handle. You'll improve with every attempt.

First, you'll create a rubber ball and create a proper animation hierarchy for it. Then, you'll add cartoonish movement to accentuate some principles of the animation techniques discussed at the end of the ultra-fabulous Chapter 1, "Introduction to Computer Graphics and 3D."

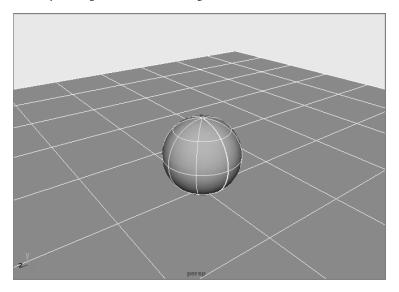
Visit the book's web page for a video tutorial on this ball animation exercise.

Creating a Cartoon Ball

First you need to create the ball, as well as the project for this exercise. Follow these steps:

- 1. In a new scene, begin with a poly sphere and then create a poly plane. Scale the plane up to be the ground plane.
- 2. Press 5 for Shaded mode.
- 3. Move the sphere 1.0 unit up in the Y-axis so that it's resting on the ground and not halfway through it, as shown in Figure 8.1.





4. Choose Modify → Freeze Transformations to set the ball's resting height to **0**, as opposed to 1. This action sets the ball's Translate attribute to 0, effectively resetting

- the object's values. This is called *freezing the transforms*. This is useful when you position, scale, and orient an object and need to set its new location, orientation, and size as the beginning state.
- 5. Choose File → Project Window to create a new project. Call the project **Bouncing Ball** and place it in the same parent folder as your Solar_System project folder. Choose Edit → Reset Settings to create the necessary folders in your project and then click Accept. Save the scene file into that project.

Setting Up the Hierarchy

First you'll set up the ball with three null nodes above it, listed here from the top parent node down: translate, scale, rotate. All the animation will be placed on these three nodes and not the sphere itself. This will allow you to easily animate the ball bouncing, squashing, stretching, and moving forward in space.

- 1. Select the sphere and press Ctrl+G to create the first group. In the Outliner, call this new group rotate.
- 2. With the rotate node selected, press Ctrl+G to create the scale group, and name it accordingly.
- 3. With the scale group selected, press Ctrl+G one last time to create the **translate** group and name it accordingly. Figure 8.2 shows the hierarchy.

As you animate, you'll quickly see why you've set up a hierarchy for the ball, instead of just putting keys on the sphere.

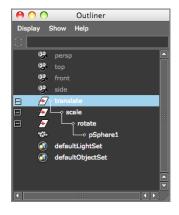


Figure 8.2 The ball's hierarchy

Animating the Ball

Your next step is to keyframe the positions of the ball using the nodes above the sphere. You'll start with the gross animation, which is the overall movement scheme, a.k.a. blocking. You'll move the ball up and down to begin its choreography in these steps:

- 1. Press W to open the Translate tool, select the translate node, and move it up to the top of the frame, say about 10 units up in the Y-axis and 8 units back in the X-axis at (-8,10,0). Place the camera so that you'll have some room to work in the frame.
- 2. Instead of selecting the Translate attributes in the Channel Box and RMB+clicking for the Key Selected command, you'll use hotkeys this time.

Press Shift+W to set keyframes on Translate X, Translate Y, and Translate Z at frame 1 for the top node of the ball (named translate). To make sure your scene is set up properly, set your animation speed to 30fps by choosing Window > Settings/Preferences → Preferences to open the Preferences window or by clicking

- the Animation Preferences button () next to the Auto Keyframe button. In the Settings category of the Preferences window, set Time to NTSC (30fps). A frame range of 1 to 120 is good for now. Figure 8.3 shows the ball's start position.
- 3. Click the Auto Keyframe button () to turn it on; it turns red. Auto Keyframe automatically sets a keyframe at the current time for any attribute that has changed since its last keyframe for the selected object or node.

For the Auto Keyframe feature to work, you first have to set an initial keyframe manually for each of the attributes you want to animate.

- 4. Disregarding any specific timing, go to frame 10 and move the ball down in the Y-axis until it's about one-quarter through the ground plane. Because you'll be creating squash and stretch for this cartoon ball (see Chapter 1 for a brief explanation), you need to send the ball through the ground a little bit. Then, move the ball about 3 units to the right, to about (-5,-0.4,0). The Auto Keyframe feature sets a keyframe in the X- and Y-axes at frame 10. Remember, this is all on the translate node, not the sphere.
- 5. Move to frame 20 and raise the ball back up to about half of its original height and to the right about 2.5 units (-2.5,4,0). Auto Keyframe sets X and Y Translation keyframes at frame 20 and will continue to set keyframes for the ball as you animate.
- 6. At frame 30, place the ball back down a little less than one-quarter of the way through the ground and about 2 units to the right, at about (-0.5, -0.3, 0).
- 7. At frame 40, place the ball back up in the air in the Y-axis at a fraction of its original height and to the right about 1.5 units, at about (1.1,1.85,0).

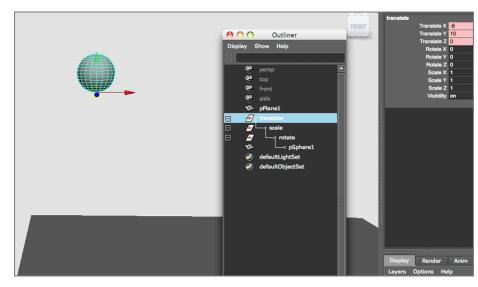


Figure 8.3 Start the ball here and set a keyframe on the translate node. 8. Repeat this procedure every 10 frames to about frame 110 so that you bounce the ball a few more times up and down and to the right (positive in the X-axis). Make sure you're decreasing the ball's height and traveling in X with each successive bounce and decreasing how much the ball passes through the ground with every landing until it rests on top of the ground plane. Open the Graph Editor for a peek into the ball's animation curves (see Figure 8.4). (Choose Window → Animation Editors → Graph Editor.)

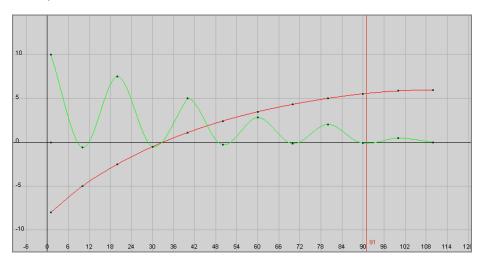


Figure 8.4 The Graph Editor curves for the ball's

translate node

By holding down the Shift key as you pressed W in step 2, you set a keyframe for Translate. Likewise, you can keyframe Rotation and Scale with hotkeys. Here's a summary of the keystrokes for setting keyframes:

Shift+W Sets a keyframe for the selection's position in all three axes at the current time

Shift+E Sets a keyframe for the selection's rotation in all three axes at the current time

Shift+R Sets a keyframe for the selection's scale in all three axes at the current time

You'll resume this exercise after a look at the Graph Editor.

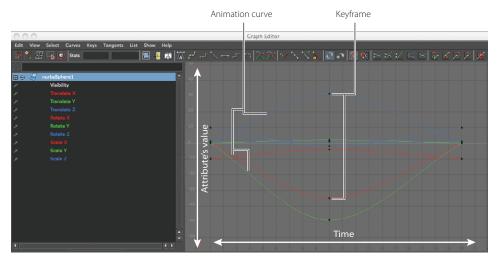
The Graph Editor

To use the Maya Graph Editor, select Window → Animation Editors → Graph Editor. It's an unbelievably powerful tool for the animator (see Figure 8.5) to edit keyframes in animation.

Every movement that is set in Maya generates a graph of value versus time, and the Graph Editor gives you direct access to those curves for fine-tuning your animation. The Graph Editor displays animation curves as value versus time, with value running vertically and time horizontally. Keyframes are represented on the curves as points that can be freely moved to adjust timing or value. This window is truly an animator's best friend. Move a keyframe in time to the right (to be later in the timeline), for example, to slow the action. Move the same keyframe to the left (to be earlier in the timeline) to speed up the action.

The Graph Editor is divided into two sections. The left column, which is much like the Outliner, displays the selected objects and their hierarchy with a listing of their animated channels or attributes. By default, all of an object's keyframed channels are displayed as colored curves in the display to the right of the list. However, by selecting an object or an object's channel in the list, you can isolate only those curves that you want to see.

Figure 8.5 The Graph Editor



Reading the Curves in the Graph Editor

Using the Graph Editor to read animation curves, you can judge an object's direction, speed, acceleration, and timing. You'll invariably come across problems and issues with your animation that require a careful review of their curves. Here are a couple of key concepts to keep in mind.

First, the curves in the Graph Editor are like the NURBS curves you've modeled with so far. This time, points on an animation curve represent keyframes, and you can control the curvature with their tangency handles. By grabbing one end of a key's handle and dragging it up or down, you adjust the curvature.

Second, the graph is a representation of an object attribute's position (vertical) over time (horizontal). So, not only does the placement of the keys on the curve make a big difference, so does the shape of the curve itself. Here is a quick primer on how to read a curve in the Graph Editor and, hence, how to edit it.

In Figure 8.6, the object's Translate X attribute is being animated. At the beginning, the curve quickly begins to move positively (that is, to the right) in the Z-axis. The object shoots off to the right and comes to an *ease-out*, where it decelerates to a stop. The stop is signified by the flat part of the curve at the first keyframe at frame 41. The object then quickly accelerates in the negative *X* direction (left) and maintains a fairly even speed until it hits frame 62, where it suddenly changes direction and goes back right for about 45 frames. It then slowly decelerates to a full stop in an ease-out.

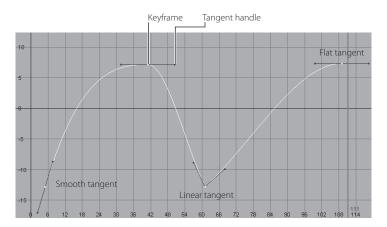


Figure 8.6 An animation curve

Consider a single object in motion. The shape of the curve in the Graph Editor defines how the object moves. The object shown in Figure 8.7 is moving in a steady manner in one direction.

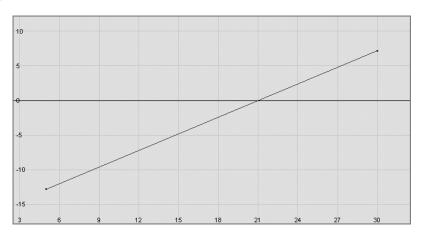
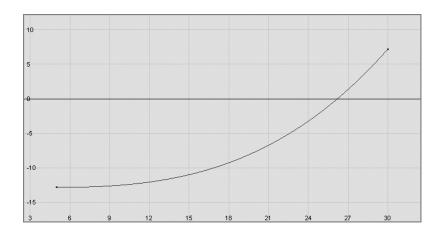


Figure 8.7 Linear movement

Figure 8.8 shows the object slowly accelerating toward frame 30, where it suddenly comes to a stop. If there is nothing beyond the end of the curve, there is no motion. The one exception deals with the *infinity* of curves, which is discussed shortly.

Figure 8.8

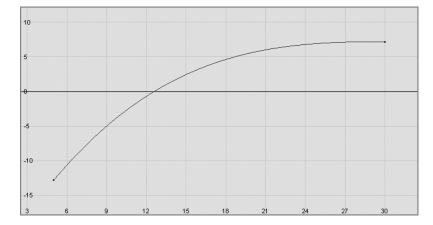
Acceleration
(ease-in)



The object in Figure 8.9 begins moving immediately and comes to a slow stop by frame 27, where the curve first becomes flat.

Figure 8.9

Deceleration
(ease-out)



Cartoon Ball

Now, let's apply what you've learned about the Graph Editor to the bouncing ball. Follow these steps:

- 1. Open the Graph Editor and look at the ball's animation curves. They should be similar to the curves shown earlier in Figure 8.4.
- 2. Notice how only the *X* and *Y*-axes' translates have curves, and yet Translate Z has a single keyframe but no curve. It's from the initial position keyframe you set at frame 1. Because you've moved the sphere only in the *X* and *Y*-axes, Auto Keyframe hasn't set any keys in the *Z*-axis.

- 3. Play back the animation and see how it feels. Be sure to open the Animation Preferences window. Click the Animation Preferences icon () to set the playback speed to Real-Time (30fps). You'll find this icon in the Playback section in the Time Slider category.
- 4. Timing is the main issue now, so you want to focus on how fast the ball bounces and move keyframes to tweak the animation. To move a keyframe, you would select it in the Graph Editor, then press W for the Move tool, and finally MMB+click to move it. But before you move any keys, do the following:
 - Watch the motion, and you'll see that the ball is falling too fast initially, although the second and third bounces should look fine.
 - To fix the timing, move the keyframes in the Graph Editor. For the *X* and Y-axes, select the keyframes at frame 10 and all the others beyond on both curves. Move them all back two frames. (See Figure 8.10.)

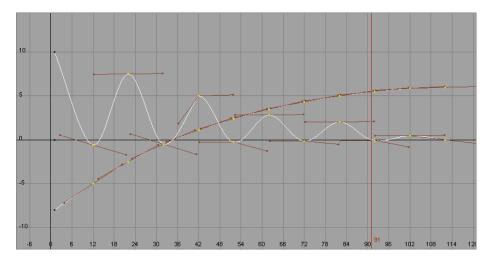


Figure 8.10

Move all the keyframes for both curves to the right to slow the initial fall by two frames, but leave the timing the same for the rest.

As the ball's bounce decays over time, it goes up less but still takes the same amount of time (10 frames) to go up the lesser distance. For better timing, adjust the last few bounces to occur faster. Select the keys on the last three bounces and move them, one by one, a frame or two to the left to decrease the time on the last short bounces. (See Figure 8.11.)

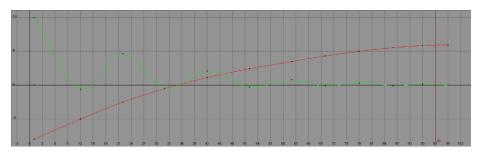


Figure 8.11

Move the keys to make the final short bounces quicker and the bounce height feel right.

To move a key in the Graph Editor, press W to open the Move tool, MMB+click, and drag the cursor in the Graph Editor window. Press the Shift key and drag the cursor left and right or up and down to lock the movement to either horizontal or vertical to make it easier to control.

Understanding Timing

In animation, timing is all about getting the keyframes in the proper order. Judging the speed of an object in animation is critical to getting it to look right, and that comes down to timing. Download the file ball_v01.mb from the Bouncing_Ball project on the companion web page, www.sybex.com/go/introducingmaya2015, to get to this point.

When you play back the animation, it seems a bit unrealistic, as if it's rising and falling on a wave as opposed to really hitting the ground and bouncing back. The problem with the animation is that the ball eases in and out as it rises and falls. By default, setting a key in Maya sets the keyframes to have an ease-in and ease-out in their curves, meaning their curves are smooth like a NURBS curve. For a more natural motion, you need to accelerate the ball as it falls with a sharp valley in the curve, and you need to decelerate it as it rises with smooth peaks. Follow these steps:

- 1. In the Graph Editor, select the Translate Y entry in the left panel of the window to isolate your view to just that curve in the editor panel on the right. Select all the landing keyframes (the ones in the valleys of the curve) and change their interpolation from smooth to linear by clicking the Linear Tangents button ().
- 2. Likewise, select all the peak keyframes at the ball's rise and change their tangents to flat by clicking the Flat Tangents icon () to make the animation curve like the one shown in Figure 8.12.

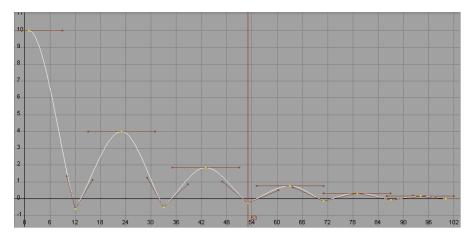


Figure 8.12 The adjusted timing of the bounce

3. When you play back the animation, the ball seems to be moving more realistically. If you need to, adjust the keys a bit more to get the timing to feel right to you, before you move on to squash and stretch and rotation.

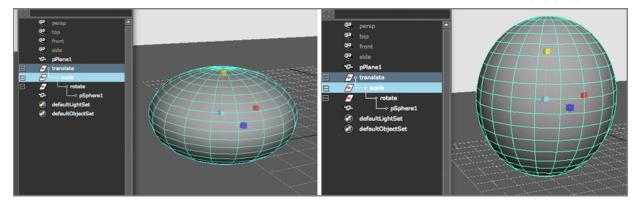
Squash and Stretch

The concept of *squash and stretch* has been an animation staple for as long as animation has existed. It's a way to convey the weight of an object by deforming it to react (usually in an exaggerated way) to gravity and motion. You can do this by scaling your object.

Download the file ball_v02.mb from the Bouncing_Ball project on the book's web page and follow these steps:

- 1. Select the scale node (not the sphere!) and select Modify → Center Pivot. This places the scale pivot point in the middle of the ball.
- 2. At frame 9, press Shift+R to set initial scale keyframes on the scale node of the ball, a few frames before the ball impacts the ground.
- 3. To initiate squash and stretch, go to frame 12, where the ball hits the floor the first time. With the scale node selected, press R to open the Scale tool; scale the ball down in the Y-axis until it no longer goes through the floor (about **0.6**), as shown in the image on the left in Figure 8.13. Set a keyframe for scale by pressing Shift+R.
- 4. Move ahead in the animation about three frames to frame 15. Scale the ball up in the Y-axis slightly past normal to stretch it up (about 1.15) immediately after its bounce, as shown in the image on the right in Figure 8.13. Three frames later, at frame 18, set the *Y*-axis scale back to 1 to return the ball to its regular shape.

Figure 8.13 Squashing and stretching the ball to react to bouncing on the floor



- 5. Scrub your animation, and you should see the ball begin stretching even before it hits the ground. That's a bit too much exaggeration, so open the Graph Editor and move the Y-axis scale key from 9 to 11. Now, the ball squashes when it hits the floor and stretches as it bounces up.
- 6. Repeat this procedure for the remaining bounces, squashing the ball as it hits the floor and stretching it as it bounces up. Remember to decay the scale factor as the ball's bouncing decays to a stop; the final bounce or two should have very little squash and stretch, if any.

Download the file ball_v03.mb from the Bouncing Ball project on the book's web page to get to this point.

Rotation

Let's add some roll to the ball in these steps:

- 1. Select the ball's rotate node and select Modify → Center Pivot to set that node's pivot at the center of the ball.
- 2. At frame 1, press Shift+E to set keys for rotation at (0,0,0).
- 3. Scrub to the end of your animation (frame 100 in this example) and set a value of -480 for Rotate Z in the Channel Box, as shown in Figure 8.14.

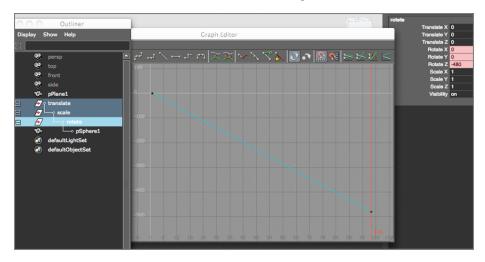


Figure 8.14 Setting a roll for the ball

> 4. Open the Graph Editor to see the rotation curve on the ball's rotate node. It's a linear (straight) line angled down from 0 to -480. You need the rotation to slow to a stop at the end of the animation, so select the final keyframe and click **—** to make it a flat tangent.

Load the file ball_v04.mb from the Bouncing_Ball project from the companion web page to see an example of the finished bouncing ball. Although the bouncing of this ball looks okay, it could definitely use some finesse, a little timing change, and so on. Open the file, open the Graph Editor, and edit the file to get a feel for animating it in your own style.

Throwing an Axe

This next project will exercise your use of hierarchies and introduce you to creating and refining motion to achieve proper animation for a more complex scene than the bouncing ball. The workflow is simple but standard for properly setting up a scene for animation, also known as *rigging*. First you'll model an axe and a target, and then you'll set up the grouping and pivots for how you want to animate. Then you'll throw your axe!

Why won't you throw the NURBS axe you've already created and textured? Because later in this chapter, you'll need it for an exercise on importing and replacing an object in Maya while keeping the animation intact. You'll see a video tutorial on the book's website for the axe throwing exercise.

The Preproduction Process

To begin the animation right away, you'll use a basic axe and bull's-eye target to focus on animation and technique.

Create a new project: Choose File → Project Window and click the New button. Place this project in the same folder or drive as your other projects, call it **Axe**, and click Accept. Click the Animation Preferences button, and set the frames per second to 30fps.

What separates good animation from bad animation is the feeling of weight that the audience infers from the animation. People instinctively understand how nature works in motion. You see an object in motion, how it moves, and how it affects its surroundings. From that, you can feel the essence of its motion, with its weight making a distinct impression on you. As it pertains to animation, that essence is simply called *weight*. A good feeling of weight in animation depends on timing and follow-through, which require practice.

It's a good idea first to try an action you want to animate. It may upset the cat if you grab a real axe and start throwing it around your house, but you can take a pen, remove its cap, and lob it across the room. Notice how it arcs through the air, how it spins around its center of balance, and how it hits its mark. Just try not to take out anyone's eye with the pen. According to some Internet research, the perfect axe throw should contain as few spins as possible.

Animating the Axe: Keyframing Gross Animation

The first step is to keyframe the positions of the axe, starting with its gross animation that is, the movement from one end of the axe's trajectory to the other.

Setting Initial Keyframes

Load the scene file axe_v1.mb in the Scenes folder of the Axe project from the companion web page and follow these steps:

1. Select the axe's top group node. To make selecting groups such as this easier, display the object's selection handle. To do so, select the axe's top node and choose Display → Transform Display → Selection Handles. Doing this displays a small cross, called a selection handle (+), at the axe's pivot point. You need only select this selection handle to select the top node of the axe.

You can turn on selection handles for practically any object in Maya—no matter where it is in a group's hierarchy—whether it's a child or a parent. Top nodes display as a hollow cross when enabled, and any child nodes appear as a regular cross (+) when selection handles are turned on.

- 2. With the axe selected, go to frame 1 and set a keyframe for the rotation and translation.
- 3. Hold down the Shift key, press W for the axe's translation keyframes, and then press E for the axe's rotation keyframes for its start position.

Creating Anticipation

Instead of the axe just flying through the air toward the target, you'll animate the axe moving back first to create *anticipation*, as if an invisible arm were pulling the axe back before throwing it. Follow these steps:

- 1. Go to frame 15.
- 2. Move the axe back in the X-axis about 8 units and rotate it counterclockwise about 45 degrees.
- 3. The Auto Keyframe feature sets keyframes for the position and new rotation at frame 15.

Because you've moved the axe back only in the X-axis and made the rotation only on the Z-axis, Auto Keyframe sets keyframes only for Translate X and Rotate Z. The other position and rotation axes aren't auto-keyframed because their values didn't change. Take note of this fact.

4. Scrub through the animation and notice how the axe moves back in anticipation.

Auto Keyframe inserts a keyframe at the current time for the selected object's changed attributes only.

5. Go to frame 40 and move the axe so that its blade cuts into the center of the target.

Notice that you have to move the axe in the *X*- and *Y*-axes, whereas before you had to move it back only in the *X*-axis to create anticipation. This is because the axis of motion for the axe rotates along with the axe. This is an example of the *Local axis*. The Local axis for any given object shifts according to the object's orientation. Because you angled the axe back about 45 degrees, its Local axis rotated back the same amount.

The file axe_v2.mb in the Scenes folder of the Axe project from the companion web page will catch you up to this point in the animation.

This last step reveals a problem with the animation. If you scrub your animation now, you'll notice that the axe's movement back is different from before, setting a keyframe at frame 40. Why intentionally create a problem like this? Troubleshooting problems in a scene is vital to getting good as a CG animator, so the more you learn how to diagnose problems, the easier production will become.

This is because of the Auto Keyframe feature. At frame 1, you set an initial keyframe for the *X*-, *Y*-, and *Z*-axes of translation and rotation. Then, at frame 15, you moved the axe back in the *X*-axis only (in addition to rotating it in the *Z*-axis only).

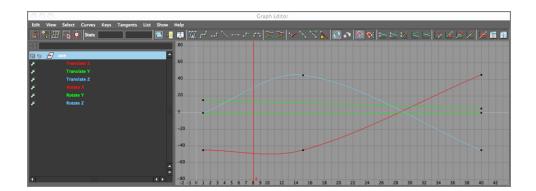
Auto Keyframe set a keyframe for Translate X at frame 15. At frame 40, you moved the axe in *both* the *X*- and *Y*-axes to strike the target. Auto Keyframe set a keyframe at 40 for Translate X and Translate Y. Because the last keyframe for Translate Y was set at 1 and not at 15 as in the case of Translate X, there is now a bobble in the *Y* position of the axe between frames 1 and 15.

With the axe selected, open the Graph Editor (choose Window → Animation Editors → Graph Editor) to see what's happening. You should see red, green, and blue line segments running up and down and left and right. You'll probably have to zoom your view to something more intelligible. By using the Alt key (or the Alt/Option key on a Mac) and mouse-button combinations, you can navigate the Graph Editor much as you can any of the modeling windows.

The hotkeys A and F also work in the Graph Editor. Click anywhere in the Graph Editor window to make sure it's the active window and press A to zoom all your curves into view. Your window should look something like Figure 8.15.

The curves in the Graph Editor represent the values of the axe's position and rotation at any given time. The X-, Y-, Z-axes are in their representative red, green, or blue color, and the specific attributes are listed much as they are in the Outliner in the left column. Selecting an object or an attribute on the left displays its curves on the right.

Figure 8.15 The Graph Editor displays the axe's animation curves.



You should also notice that the curves are all at different scales. The three rotate curves range in value from about -45 to 45, the Translate Y curve ranges from about 15 to 5, and Translate Z looks flat in the Graph Editor. It's tough to edit a curve with low values and still be able to see the timings of a larger value curve.

You can select the specific attribute and zoom in on its curve to see it better, or you can normalize the curves so that you can see them all in one view. Click the Enable Normalized Curve Display icon in the top icon bar of the Graph Editor (). The values don't change, but the curves now display in a better scale in relation to each other so you can see them all together.

Figure 8.16 shows the Graph Editor from Figure 8.15 after the curves have been normalized. Keep in mind that this doesn't change the animation in the slightest. All it does is allow you to see all the curves and their relative motion. You can denormalize the view by clicking the Disable Normalized Curve Display icon in the Graph Editor ().

ᄺᇸᄶᆈᅪᅩᅩᄼᆛᄶᄶᄶᄼᄼᄼᇸᄝᇬᇒᇸᅜᇎᄯᅜᅜᅜᄺ

Figure 8.16 The normalized view in the Graph Editor lets you see all the curves of an animation together in the same scale.

Notice that the curve for Translate Y has keyframes only at frames 1 and 40. The animation dips in the first 15 frames because there is no keyframe at frame 15 like there is for Translate Z. That dip wasn't there before you set the end keyframe at frame 40.

Continue the exercise by fixing this issue:

- 1. Move the first keyframe of Translate Y from frame 1 to frame 15 to fix the dip.
 - Press W to activate the Move tool in Maya, or click the Move Nearest Picked Key Tool icon () in the Graph Editor.
 - Click the Time Snap On/Off icon (\sum_{\infty}) to toggle it on.
 - Select the offending Translate Y keyframe at frame 1, and MMB+click and drag it to the right until it's at frame 15.

Scrub your animation, and the backward movement looks like it did before.

The axe now needs an arc on its way to the target.

- 2. Go to the middle of the axe's flight, frame 27.
- 3. Move the axe up in the *Y*-axis a bit using the green handle of the Tool manipulator.

If the axe is slightly rotated in frame, Auto Keyframe can set a key for both Translate Y and Translate X, although you were perhaps expecting only a key in Translate Y. Because the Move tool is on the axe's Local axis and because the axe was slightly rotated at frame 27, there is a change in the *Y* and *X* positions in the World axis, which is the axis represented in the Graph Editor.

- 4. Select the Translate X key at frame 27 in the Graph Editor, if one was created, and press Delete to delete it.
- 5. Now you'll add a full spin to the axe to give the animation more reality and life. You can spin it in one of two ways:
 - Go to frame 40, select the axe, and rotate it clockwise a full 360 degrees positive. Auto Keyframe enters a new rotation value at frame 40, overwriting the old value. You should see the Rotate Z curve angle down steeply as soon as you let go of the Rotate manipulator.
 - In the Graph Editor, make sure you're at frame 40, grab the last keyframe on the Rotate Z curve, and MMB+click and drag it down, probably past the lower limit of the window. If you keep the middle mouse button pressed as you move the mouse, the keyframe keeps moving as you move the mouse, even if the keyframe has left the visible bounds of the Graph Editor.

If you hold down Shift as you MMB+click and drag the keyframe to move it in the Graph Editor, the keyframe will move in only one axis (up or down, left or right).

By moving the keyframe down, you change the Rotate Z value to a lower number, which spins the axe clockwise. Before you try that, though, move your Graph Editor

- window so you can see the axe in the Perspective window. As you move the Rotate Z keyframe down in the Graph Editor, you see the axe rotate interactively. Move the keyframe down until the axe does a full spin.
- 6. Play back the animation by clicking the Play button in the playback controls. If your animation looks blazingly fast, open the Animation Preferences window by clicking its icon (and set Playback Speed to Real-Time (30fps).

Now, when you play back the animation, it should look slow. Maya is playing the scene back in real time, but even at 30fps, the scene plays back slowly, and this means that the animation of the axe is too slow.

7. All you need to do is tinker in the Graph Editor a bit to get the right timing. For a good result in timing, move the first set of keyframes from 15 to 13. Then, grab the Translate Y keyframe at frame 27 and move it to 19. Finally, grab the keyframes at frame 40 and move them all back to frame 25. Play back the scene.

Changing the playback speed of an animation through the Animation Preferences window doesn't alter the timing of your animation. It only changes the speed at which Maya plays the animation back to you in its windows. To change the playback speed, choose Window → Setting/Preferences → Preferences to open the Preferences window, choose Settings → Working Units, and select the proper setting.

Adding Follow-Through

Load the axe_v3.mb file from the Axe project on the book's web page or continue with your own file.

The axe is missing weight. You can add some finesse to the scene using follow-through and secondary motion to give more weight to the scene.

In the axe scene, follow-through motion is the axe blade driving farther into the target a little beyond its initial impact. Secondary motion is the recoil in the target as the momentum of the axe transfers into it. As you increase the amount of follow-through and secondary motion, you increase the axe's implied weight. You must, however, walk a fine line; you don't want to go too far with follow-through or secondary motion. Follow these steps:

- 1. Select the axe in the scene using its selection handle and open the Graph Editor.
- 2. Because you'll add three frames to the end of this animation for follow-through, go to frame 28 (25 is the end of the current animation).
- 3. In the Perspective window, rotate the axe another 1.5 degrees in the Z-axis.
- 4. Rotating the axe in step 3 moves the axe's blade down a bit in the Y-axis. To bring the axe back up close to where it was before the extra rotation, move the axe up slightly

using the Translate Y manipulator handle. This also digs the axe into the target a little more. You'll see a keyframe for Translate Y and most probably for Translate X, as well as for Rotate Z.

If you play back the animation, the follow-through doesn't look good. The axe hits the target and then digs into it as if the action were done in two separate moves by two different animators who never talked to each other. You need to smooth out the transition from the axe strike and its follow-through in the Graph Editor.

5. Highlight the Rotate Z attribute in the Graph Editor to get rid of the other curves in the window. Figure 8.17 shows the Rotate Z curve of the axe after the follow-through animation is added.

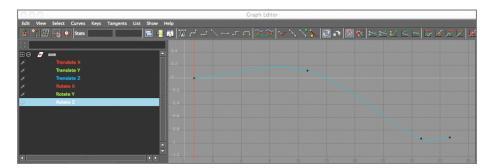


Figure 8.17

The normalized Rotate Z curve of the axe after the follow-through animation

6. Focus on the last three frames of the curve and zoom into that range only. The curve, as it is now, dips down past where it should and recoils back up a small amount.

When you set keyframes for the axe, you create Bézier splines animation curves in the Graph Editor. These curves stay as smooth as possible from beginning to end. When you set the new keyframe, rotating the axe about 1.5 more degrees for follow-through, the animation curve responds by creating a dip, as shown in Figure 8.17, to keep the whole curve as smooth as possible.

SECONDARY MOTION AND FOLLOW-THROUGH

Secondary motion in animation comprises all the little things in a scene that move because something else in the scene is moving. For example, when a superhero jumps from a tall building and his or her cape flutters in the wind, the cape's undulation is secondary motion.

Follow-through is the action in animation that immediately follows an object's or a character's main action. For example, after the superhero lands from their jump, his or her knees buckle a little, and the superhero bends at the waist, reacting to his or her weight as it settles, creating follow-through.

The axe needs to hit the target with force and dig its way in, slowly coming to a stop. You need to adjust the curvature of the keyframes at frame 25 by using the keyframe's tangents. Tangents are handles that change the amount of curvature influence of a point on a b-spline (Bézier spline). Selecting the keyframe in question reveals its tangents, as shown in Figure 8.18.

- 7. Select the Out tangent (handle on the right side of the key) for the Rotate Z attribute's key at frame 25 and MMB+click and drag it up to get rid of the dip. Notice that the tangency for the In tangent (handle on the left side of the key) also changes.
- 8. Press Z to undo your change. You need to break the tangent handles so that one doesn't disturb the other.
- 9. Select the Out handle and click the Break Tangents icon () to break the tangent.
- 10. Move the handle up to get rid of the dip so that the curve segment from frame 25 to frame 28 is a straight line, angled down. Figure 8.19 is zoomed into this segment of the curve after it's been fixed.

-411



-412 -413 -414

Figure 8.18

The tangent handles of a keyframe. The handle to the left of the keyframe is the In tangent, and the handle to the right is its Out tangent.

Figure 8.19

Zoomed into the end segment of the Rotate Z animation curve after the dip is fixed

Now, to get the axe to stop slowly as it digs into the target, you need to curve that end segment of the Rotate Z curve to flatten it out.

11. Grab the last frame to reveal its handles. You can manually move the In handle to make it horizontal, or you can click the Flat Tangents icon (on the left side of the icon bar, under the menus in the Graph Editor.

The curve's final segment for Rotate Z should now look like Figure 8.20.

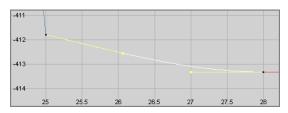


Figure 8.20

Zoomed into the end segment of the Rotate Z animation curve. Notice how the curve now smoothly comes to a stop by flattening out.

- 12. Adjust the keyframe tangents similarly for the axe's Translate Y and Translate X curves, as shown in Figure 8.21.
- 13. Play back the animation, and you should see the axe impact the target and sink into it a bit for its follow-through.

Now, you need to polish things up more.

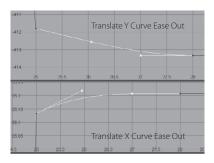


Figure 8.21
Smoothed translate curves to ease out the motion

Adding Secondary Motion

Load axe_v4.mb from the Axe project from the book's web page, or continue with your own scene file. For secondary motion, you'll move the target in reaction to the impact from the axe's momentum.

An object in motion has momentum. Momentum is calculated by multiplying the mass of an object by its velocity. So, the heavier and faster an object is, the more momentum it has. When two objects collide, some or all momentum transfers from one object to the other.

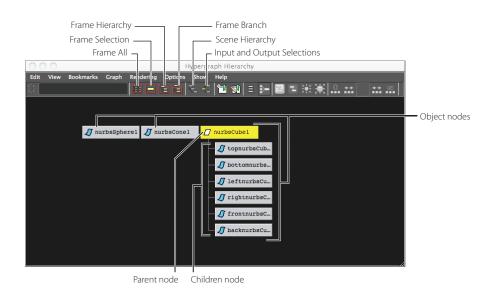
In the axe scene's impact, the axe lodges in the target, and its momentum is almost fully transferred to the target. But because the target is more massive than the axe, the target moves only slightly in reaction. The more you make the target recoil, the heavier the axe will seem.

First, group the axe's parent node under the target's parent node by grouping the axe under the target; when you move the target to recoil, it will keep the axe lodged in it. You will use the Hypergraph instead of the Outliner, so let's take a quick look at its interface first.

The Hypergraph Explained

The Hypergraph: Hierarchy (referred to as just the Hypergraph in this book) displays all the objects in your scene in a graphical layout similar to a flowchart (see Figure 8.22). Select Window → Hypergraph: Hierarchy to see the relationships between objects in your scene more directly. This window will perhaps be somewhat more difficult for a novice to decipher, but it affords you great control over object interconnectivity, hierarchy, and input and output connections. The Hypergraph: Connections window is technically called the Hypergraph window, but it shows you the interconnections of attributes among nodes as opposed to the node hierarchy in the scene.

Navigating the Hypergraph is the same as navigating any Modeling window, using the familiar Alt key and mouse combinations for tracking and zooming.



Continuing the Axe

First you need to reset the target node's Translate and Rotate attributes to **0** and its Scale attributes to **1**.

- 1. To freeze the transforms, select the parent target node and then choose Modify → Freeze Transformations.
- 2. Select Window → Hypergraph: Hierarchy. MMB+click and drag the axe node to the target node in the Hypergraph to group the axe node under the target node. This can also be done in the Outliner.
- 3. Go to frame 25, the moment of impact, and set the position and rotation keyframes on Target.
- 4. Go to frame 28, rotate the target node in the *Z*-axis about 2.5 degrees, and move it up and back slightly in the *Y* and *X*-axes, as shown in Figure 8.23.

Y . ront

- 5. Go to frame 31. Rotate the target node back to 0 in the *Z*-axis, move it down to 0 in the *Y*-axis, and move it back a bit more in the *X*-axis.
 - 6. Go to frame 35 and repeat step 4, but move it only half as much in Rotate Z, Translate X, and Translate Y.
 - 7. Go to frame 40 and repeat step 5, but move Target back only slightly in the *X*-axis.

Figure 8.23

The front-panel display of the target reacting to the impact of the axe

If you don't freeze the transforms on the target's parent node before grouping the axe under it, the axe's animation will change and yield undesirable results.

The preceding steps should give you an animation similar to <code>axe_v5.mb</code> in the Axe project from the companion web page.

Path Animation

As an alternative to keyframing the position of the axe, you can animate it on a path. Path animation allows you to assign an object to move along the course of a curve, called a *path*.

Load axe_v6.mb from the Axe project from the companion web page. This is the finished axe animation, but the translation keys have been deleted, though the rotation keys and everything else are intact. You'll replace the translation keyframes with a motion path instead. There is an arched curve in this scene called pathCurve that was already created for you using the CV curve tool. Look how nice I am to you!

- Once you open the scene (axe_v06.mb), scrub the animation and you'll see that the
 axe spins around; it and the target recoil at the moment of impact, but the axe doesn't
 actually move. There is also a curve in the scene that represents the eventual motion
 of the axe.
- 2. Select the axe node in the Outliner (Figure 8.24) and then Shift+select the path curve. In the Animation menu set, choose Animate → Motion Paths → Attach To Motion Path □.
- 3. In the option box, turn off the Follow check box.

The Follow feature orients the object on the path so that its front always points in the direction of travel. This will interfere with the axe's motion, so you'll leave it off. Click Apply, and now the axe will follow the line from end to end between frame 1

and frame 60. Of course, you have to adjust the timing to fit better, as with the original axe animation.

The file axe_path_v1.mb in the Axe project will bring you up to this point.

4. Select the axe node and open the Graph Editor to see the axe's curves. There is a purple curve called motionPath1.U Value curve that replaces the translation curves. On the left side of the Graph Editor, select the motionPath1.U Value curve to display only that. Zoom into it. (Press A to view all.)

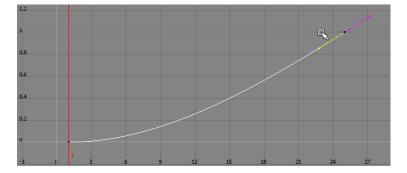


Figure 8.24

Select the Axe project's group node.

5. The curve is an ease-in and ease-out curve from 1 to 60. You need the axe to hit at frame 25, so move the end of that curve to frame 25 from frame 60. Then use the tangent handle to shape the end of the curve to be more like Figure 8.25. This will add acceleration to the end of the curve for more of a punch on the hit.

Figure 8.25 Set the last key to frame 25 and reshape the curve at the end.



- 6. Scrub the animation until the axe moves all the way back (frame 4). You'll add a keyframe here to help retime the backward movement. Select the purple U Value animation curve and then select the Insert Keys tool () in the top-left corner of the Graph Editor. MMB+click the curve to create a key at frame 4. (You can MMB+click and then drag the cursor to place the key precisely at frame 4 before releasing the mouse button.)
- 7. Move this new keyframe to frame 7. Scrub the animation, and the timing is just about right. You'll have to adjust the tangents a bit to make the axe move more like before, but the movement is essentially there with path animation.

The file axe_path_v2.mb in the Axe project will bring you up to this point.

Path animation is extremely useful, especially for animating an object along a particular course. By adjusting the resulting animation curve in the Graph Editor, you can easily readjust the timing of the path animation, and you can always adjust the shape of the path curve itself to change the motion of the object. A good path animation exercise is to reanimate the Solar_System exercise with paths instead of the keyframes you set on the rotations.

Replacing an Object

It's common practice to animate a proxy object—a simple stand-in model that you later replace. The next exercise will show you how to replace the axe you already animated with a fancier NURBS model and how to copy an animation from one object to another.

Replacing the Axe

Load the scene file axe_v5.mb from the book's web page. Now, follow these steps:

1. Choose File → Import.

2. Locate and import the Axe_Replace.ma scene file from the Scenes folder of the Axe project from the companion web page. The new axe appears at the origin in your scene.

Transferring Animation

Transferring all the properties and actions of the original axe to a new axe requires some setup. Follow these steps:

- 1. Move the pivot on the new axe to the same relative position as the pivot on the original animated axe (up toward the top and a little out front of the handle, just under the blade). This ensures that the new axe has the same spin as the old axe. Otherwise, the animation won't look right when transferred to the new axe. Figure 8.26 shows the pivot's location.
- 2. Use grid snap to place the top node of the new axe at the origin.
- 3. Choose Display → Transform Display → Selection Handles to turn on the selection handle of the new axe.
- 4. Go to frame 1. Select the original axe's axe node, open the Graph Editor, and choose Edit → Copy.
- 5. Select the new axe to display its curves in the Graph Editor. It has no curves to display yet. With the new Axe node selected in the Graph Editor, choose Edit → Paste. As shown in Figure 8.27, the new axe is slightly offset from the original axe.

When you copy and paste curves in the Graph Editor, make sure you're on the first frame of the animation. Pasting curves places them at the current frame. Because the animation of the original axe started at frame 1, make sure you're at frame 1 when you paste the curves to the new axe.

- 6. You have to move the new axe to match the original, but because it's already animated, you'll move it using the curves in the Graph Editor. With the new axe selected, in the Graph Editor select the Translate Y curve; move it down to match the height of the original axe.
- 7. Scrub the animation to see that the new axe has the same animation except at the end when it hits the target. Remember that you grouped the original axe under the target node for follow-through animation. Place the new axe under this node as well.

The file axe_v7.mb in the Axe project from the book's web page has the new axe imported and all the animation copied to get you caught up to this point.

8. After you scrub the animation and make sure the new axe animates properly, select the original axe's node and delete it.

Figure 8.26 Place the pivot point on the new

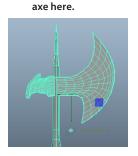
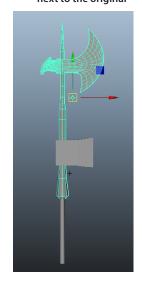


Figure 8.27 The new axe placed next to the original



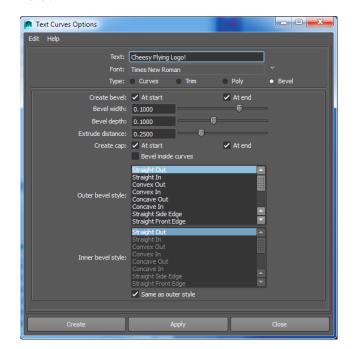
Animating Flying Text

It's inevitable. Sooner or later, you will need to animate a flying logo or flying text. Animating flying text—at least, the way you'll do it here—can show you how to animate pretty much anything that has to twist, wind, and bend along a path; this technique isn't just for text.

You'll need to create the text, so follow these steps:

1. In a new scene, select Create → Text □. In the option box, enter your text and select a font to use (see Figure 8.28). In this case, stick with Times New Roman. Set Type to Bevel.

Figure 8.28 The Text Curves **Options** box



Setting the Type attribute to Poly for your text creates curve outlines for the text and planar faces for the letters, for a flat-text effect. Setting the Type Creation option to Curves gives you just the curve outlines. Finally, using Trim to create your text makes the letters out of flat planar NURBS surfaces. However, no typographical surface history is created with text. To allow you to edit the text, you must re-create the text and/or font as needed.

- 2. Leave the rest of the creation methods at their defaults. Doing so creates the text as beveled faces to make solid text with thickness.
- 3. When you have the text, you need to create a curve for it to animate along. Using either the CV or EP Curve tool, create a winding curve like a rollercoaster for the logo, shown with the text in Figure 8.29.

Figure 8.29
Create a curve path for the text to follow.

4. Just as you did with the axe exercise, you'll assign the text object to this curve. Set your frame range to 1–100. Select the text, Shift+select the curve, and, in the Animation menu set, choose Animate \rightarrow Motion Paths \rightarrow Attach To Motion Path \square . Set Front Axis to X and Up Axis to Y, and select the Bank check box, as shown in Figure 8.30.

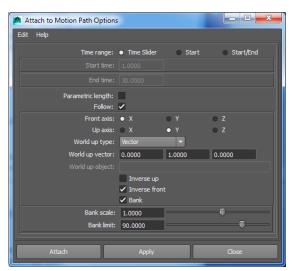
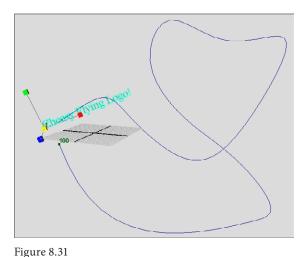


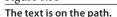
Figure 8.30
The Attach To
Motion Path
Options box

Depending on how you create your curve and text, you may need to experiment with the Front Axis and Up Axis attributes to get the text to fly the way you want.

- 5. Orbit the camera around to the other side, and you can see the text on the path, as shown in Figure 8.31. In the Attribute Editor for the motion path, notice the U Value attribute: This is the position of the text along the curve from 0 to 1. Scrub the animation, and the text should glide along the curve.
- 6. The text isn't bending along with the curve at all yet. To accomplish this, you need to add a lattice that bends the text to the curvature of the path. Select the text object, and choose Animate → Motion Paths → Flow Path Object □.
- 7. In the option box for the Flow Path Object, set Divisions: Front to 120, Up to 2, and Side to 2, and select Curve for Lattice Around. Make sure the Local Effect box is unchecked. Doing so creates a lattice that follows the curve, giving it 120 segments along the path. This lattice deforms the text as it travels along the path to the curvature of the path.
- 8. Scrub your animation, and you see a fairly strange result: Parts of the text explode out from the lattice, as shown in Figure 8.32.
- 9. The geometry is going outside the influence of the lattice. To fix the situation, select the lattice and base node; then scale up the lattice and its base node together to create a larger size of influence around the path (see Figure 8.33).

Scrub your animation to check the frame range and how well the text flies through the lattice. When the lattice and its base are large enough to handle the text along all of the path's corners and turns, *voilà*—Cheesy Flying Logo! (See Figure 8.34.)





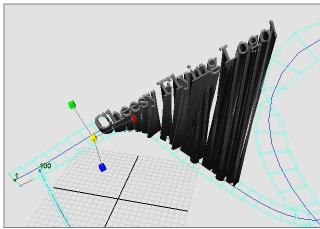


Figure 8.32 The geometry doesn't fit the lattice well. Ouch!

Figure 8.33

Scale the lattice and its base node to accommodate the flying text.

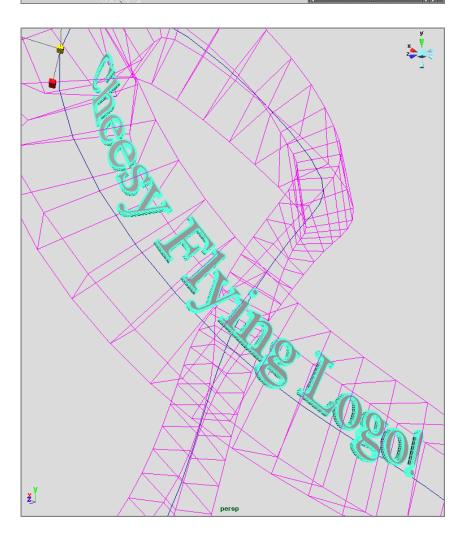


Figure 8.34
Cheesy Flying Logo! makes a nice turn.

Animating the Catapult

As an exercise in animating a system of parts of a model, you'll now animate the catapult from Chapter 4, "Beginning Polygonal Modeling." You'll turn its winch to bend back the catapult arm, which shoots the projectiles, and then you'll fire and watch the arm fly up.

Let's get acquainted with the scene file and make sure its pivots and hierarchies are set up properly. The scene file catapult_anim_v1.mb in the Catapult_Anim project from the companion web page has everything in order, although it's always good to make sure. Figure 8.35 shows the catapult with its winch selected and ready to animate.

Get a sense of timing put down for the winch first and use that to pull back the arm to fire. Follow these steps:

- 1. Select the Winch group with its selection handle. At frame 1, set a keyframe for rotation. If the selection handle isn't turned on, select Winch from the Outliner and turn on the selection handle by choosing Display → Transform Display → Selection Handles. To keep it clean as you go along, instead of pressing Shift+E to set a key for all three axes of rotation, select only the Rotate X attribute in the Channel Box, rightclick to open the context menu, and choose Key Selected. There only needs to be rotation in *X* for the winch.
- 2. Jump to frame 60. Rotate the winch backward a few times or enter -400 or so for the Rotate X attribute.
- 3. Open the Graph Editor, ease in the curve a bit, and ease out the curve a lot so that the rotation starts casually but grinds to a stop as the arm becomes more difficult to pull back.

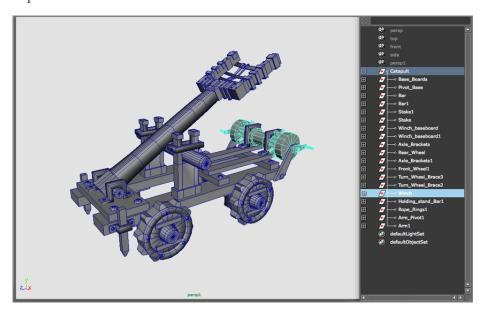


Figure 8.35 The catapult's winch is ready to animate.

Because animating a rope is a fairly advanced task, the catapult is animated without its rope; however, the principle of an imaginary rope pulling the arm down to create tension in the arm drives the animation.

- 4. To accentuate the more difficult winding at the end, add a key to the *X*-axis rotation through the Graph Editor. To do so, select the curve and click the Insert Keys Tool icon () in the upper-left corner of the Graph Editor.
- 5. MMB+click and drag to place a new keyframe positioned at frame 42. You can always drag the key back and forth on the curve to place it directly at frame 42. It may help to turn on key snapping first. (See Figure 8.36.)

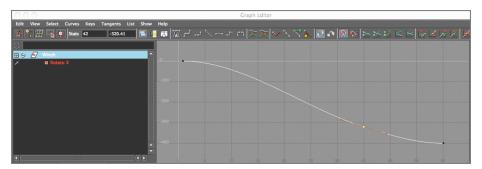


Figure 8.36
Insert a keyframe at frame 42.

6. Move that keyframe down to create a stronger ease-out for the winch. Be careful not to let the curve dip down so that the winch switches directions. Adjust the handles to smooth the curve. You can also add a little recoil to the winch by inserting a new keyframe through the Graph Editor at frame 70. (See Figure 8.37.)

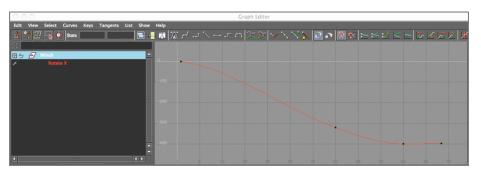


Figure 8.37

Creating a greater ease-out and adding a little recoil at the end

Animating with Deformers

It's time to animate the arm coiling back, using the winch's timing as it's driving the arm. Because the catapult's arm is supported by a brace and the whole idea of a catapult is based on tension, you have to bend the arm back as the winch pulls it.

You'll use a nonlinear deformer, just as you did in the axe head exercise in Chapter 5, but this time, you'll animate the deformer to create the bending of the catapult arm. Follow these steps:

1. Switch to the Animation menu set. Select the Arm1 group, and choose Create Deformers → Nonlinear → Bend to create a Bend deformer perpendicular to the arm. Select the deformer and rotate it to line it up with the arm, as shown in Figure 8.38.

Figure 8.38 Align the Bend deformer with the catapult arm.

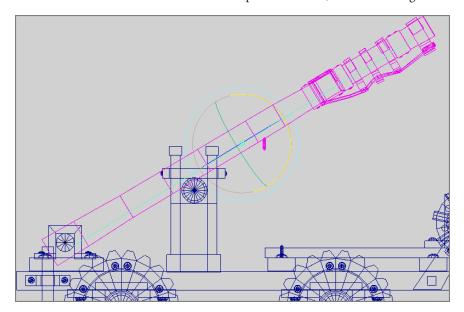
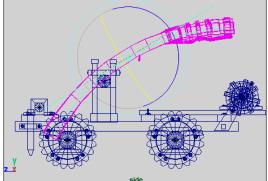


Figure 8.39 Orient the Bend deformer to bend the arm back and down.

2. With the Bend deformer selected, look in the Channel Box for bend1 under the Inputs section, and click it to expand its attributes. Try entering **0.5** for Curvature. More than likely, the catapult arm will bend sideways. Rotate the deformer so that the arm is bending back and down instead, as shown in Figure 8.39. 3. You don't want the arm's base to bend back, just the



- basket side. You want it to bend at the brace point, not in the middle where it is now. Move the deformer down the length of the arm until the middle lines up with the arm's support brace.
- 4. To prevent the bottom of the arm from bending, change the Low Bound attribute to **0**. To keep the basket from bending, set the High Bound attribute to 0.9.
- 5. Instead of trying to match the speed, ease-in, and easeout of the winch, set the gross keyframes for the arm pulling back first. Reset Curvature to **0**, and set a key for

Curvature at frame 1. (Select bend1's Curvature in the Channel Box, right-click, and choose Key Selected from the context menu.)

The Low Bound and High Bound attributes control how far up and down the deformer the object is affected. The Envelope attribute for a deformer governs how much the object is affected overall, with 0 not affecting the geometry at all.

6. Go to frame 60 and set Curvature to **0.8**. If Auto Keyframe is turned on, this sets a keyframe; otherwise, set a key manually. (See Figure 8.40.)

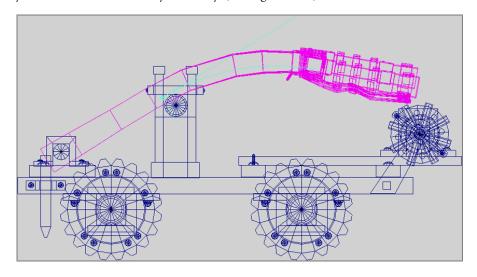


Figure 8.40 Bend the arm back at frame 60 and set

a keyframe.

- 7. If you play back the animation, notice that the way the winch winds back and the way the arm bends don't match. In the Graph Editor, you can adjust the animation curve on the Bend deformer to match the winch's curve.
- 8. Insert a key on the Curvature curve at frame 42 and move it up to match the curvature you created for the winch.
- 9. Insert a new key at frame 70 and make the arm bend back up slightly as the winch recoils. Set Curvature to about **0.79** from 0.8. (See Figure 8.41.)

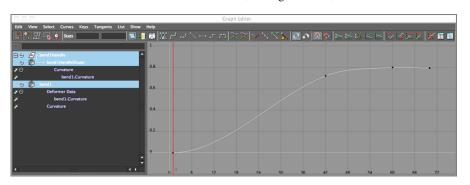
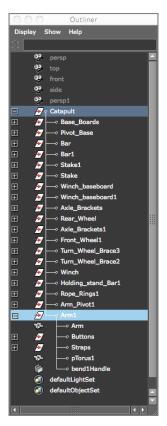


Figure 8.41

Try to match the relative curvature of the winch's animation curve with the Bend deformer's animation curve.

Figure 8.42

Group the deformer node under the Arm1 group node.

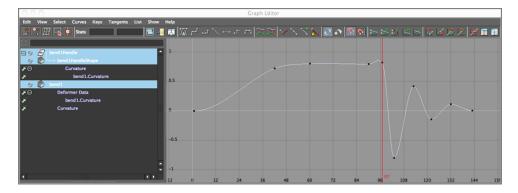


- 10. Go to frame 90 and set a key again for Curvature at
 0. Set a key at 0.82 for frame 97 to create anticipation and then keyframe at frame 103 to release the arm and fire the imaginary payload with a Curvature of -0.8.
- 11. Add some rotation to the arm for dramatic effect. At about frame 100, during the release, the arm is almost straight. Select the Arm1 group and set a rotation key on the *X*-axis. Go to frame 105 and rotate the arm 45 degrees to the left in the *X*-axis. If the starting rotation of the arm is at 30 (as it is in the sample file), set an *X*-axis rotation key of 75 at frame 105.
- 12. Notice that the arm is bending strangely now that it's being rotated. It's moving off the deformer, so its influence is changing for the worse. To fix this, go back to frame 100 and group the deformer node (called bend1Handle) under the Arm1 group, as shown in Figure 8.42. Now it rotates along with the arm, adding its own bending influence.
- 13. Work on setting keyframes on the deformer and the arm's rotation so that the arm falls back down onto the support brace and quivers until it becomes straight again. The animation curve for the Bend deformer

should look like Figure 8.43. The rotation of the arm should look like Figure 8.44. Remember to make the tangents flat on the keys where the arm bounces off the brace and at the peaks, like the ball's bounce from earlier in the chapter.

Figure 8.43

The animation curve for the arm's vibration back and forth as it comes to a rest



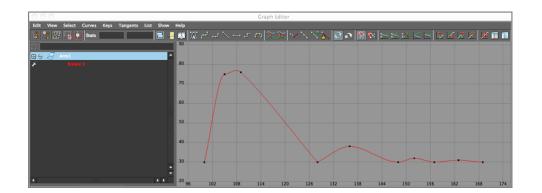


Figure 8.44

The animation curve for the arm's rotation as it heaves up and falls back down, coming to an easy rest on the brace

The file catapult_anim_v2.mb will give you a good reference to check out the timing of the arm bend and rotation. Here are some items you can animate to make this a complete animation:

- Spin the winch around as the arm releases, as if its rope is being yanked away from it.
- Animate the entire catapult rocking forward and backward as the arm releases, similar to the way a car rocks when you jump onto the hood.
- Move the catapult forward on a road, spinning its wheels as best you can to match
 the distance it travels.
- Design and build your own catapult and animate it along the same lines.

Summary

In this chapter, you began to learn the fundamentals of animating a scene. Starting with a bouncing ball, you learned how to work in the Graph Editor to set up and adjust timing as well as how to add squash and stretch to the animation. The next exercise, throwing an axe, expanded on your experience of creating timing in the Graph Editor and showed you how to add anticipation, follow-through, and secondary motion to your scene. You then learned how to animate the axe throw using path animation. You went on to learn how to replace a proxy object that is already animated with a different finished model and how to transfer the animation. Finally, you used the catapult to animate with deformers and further your experience in the Graph Editor.

Animating a "complex" system, such as a catapult, involves creating layers of animation based on facets of the mechanics of the system's movement. With the catapult, you tackled the individual parts separately and then worked to unify the animations. You'll use rigging concepts in the next chapter to automate some of that process for a

locomotive. The same is true of the Bouncing_Ball and Axe_Throwing exercises. The different needs of the animation were addressed one by one, starting with the gross animation and ending with finishing touches to add weight.

Animation is the art of observation, interpretation, and implementation. Learning to see how things move, deciphering why they move as they do, and then applying all that to your Maya scene is what animation is all about.



More Animation!

Now that you have a little more animation experience, you can get into some more involved animation practices and toolsets, exploring the principles covered in this book and its examples. Animation is a long journey, and you should use this book as a stepping-off point. For everything you're being exposed to here, there are many more techniques to discover.

Learning Outcomes: In this chapter, you will be able to

- Use hierarchies in animation tasks
- Create and manipulate a skeleton system for animation and group models to bones
- Create a walk cycle using forward kinematics for pose animation
- Discern the two ways—smooth and interactive—to bind a mesh to a skeleton in Maya
- Use inverse kinematics in a rig of a simple character
- Create a walk cycle using IK animation
- Use constraints to automate animation
- Create set-driven keys for animation rigging
- Rig a locomotive for automated animation
- Rig a simple character for basic character animation

Skeletons and Kinematics

In your body, your muscles move the bones of your skeleton, and as your bones move, parts of your body move. In CG animation, a skeleton is an armature built into a 3D model that drives the geometry when the bones are moved. You insert a skeleton into a CG model and attach or bind it to the geometry. The skeleton's bones are animated, which in turn move the parts of the geometry to which they're attached. By using a skeleton, the Autodesk® Maya® software allows bending and deformation of the attached geometry at the skeleton's joints. A skeleton is useful for character work, but skeletons have many other uses. Any time you need to drive the geometry of a model with an internal system, such as a fly-fishing line or a tree bending in the wind, you can use skeletons. You'll use skeletons to drive locomotive wheels and a simple character later in this chapter.

Skeletons and Hierarchy

Skeletons rely on hierarchies (see Figure 9.1). Bones are created in a hierarchical manner, resulting in a *root joint* that is the parent of all the joints beneath it in the hierarchy. For example, a hip joint can be the root joint of a leg skeleton system in which the knee joint is the leg's child, the ankle joint belongs to the knee, and the five toe joints are the ankle's children.

Geometry need not deform to be attached to a bone system; objects can be grouped with or under joints. They move under their parent joint and rotate around that joint's pivot as opposed to their own pivot point.

A skeleton is really just a collection of grouped and properly positioned pivot points called *joints* that you use to move your geometry. A *bone* is the length between each joint; bones show you only the skeletal system.

Inverse kinematics (IK) and forward kinematics (FK) are the methods you use to animate a skeletal system. FK rotates the bones directly at their top joint to assume poses. This method resembles *stop-motion animation*, in which a puppet, along with its underlying armature, is posed frame by frame. With FK, the animator moves the character into position by rotating the joints that run the geometry.

The rotation of a joint affects the position of the bones and joints beneath it in the hierarchy (see Figure 9.2). If you rotate the hip up, the knee and ankle swing up as if the character is kicking.

IK uses a more complex, but often easier, system of IK handles that are attached to the tip of a joint system. The corresponding base of the IK system is attached farther up the skeleton hierarchy to a joint determined to be the root of that IK segment. It need not be the root joint of the entire skeleton, though.

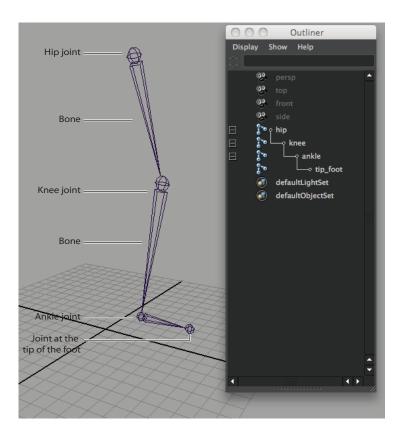


Figure 9.1 A leg skeleton and its hierarchy

The bones and joints in the IK chain are affected only by movement of the IK handle. When the handle moves, an IK solver figures out how to rotate all the joints to accommodate the new position of the IK handle.

The effect is as if someone grabbed your hand and moved it. The person holding your hand is similar to an IK handle. Moving your hand causes the bones in your arm to rotate around the shoulder, elbow, and wrist. As you can see in Figure 9.3, the animation flows up the hierarchy and is therefore called inverse kinematics.

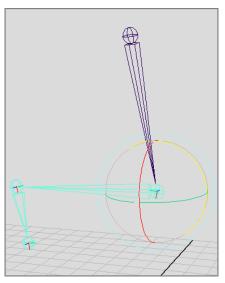
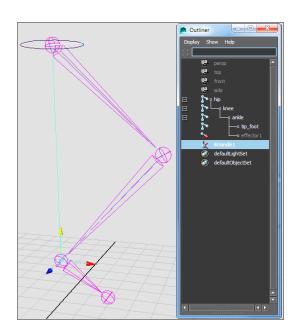


Figure 9.2 In forward kinematics, the joints are rotated directly.

Figure 9.3 In inverse kinematics, the joints rotate in response to the IK handle's position.



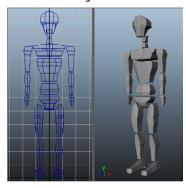
Forward Kinematics: The Block Man

To understand skeletal hierarchy, look at Figure 9.4, which shows a simple biped (twolegged) character made of primitive blocks. He's called the Block Man. (Surprise!) Each block represents part of the body, with gaps between the blocks representing points where the body pivots.

The pivot of each block is placed to represent the appropriate joint location. For example, the shin's pivot is located at the knee. Each block is grouped up the chain so that the foot moves with the shin, which moves with the thigh, which moves with the pelvis.

The hands are grouped under the arms, which are grouped under the shoulders, and so forth, down the spine to the pelvis. The head groups under the first neck block and so on down the spine to the pelvis. The pelvis is the center of the body, which is known as the root of the figure.

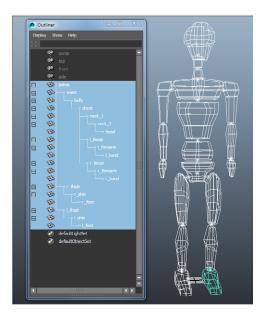
Figure 9.4 The Block Man's cubes arranged



The way this figure is grouped (see Figure 9.5) represents how the hierarchy of a character works for the most part. Each body part is attached and becomes the child of the part above it in the chain.

Load the file block_man_v02.ma from the Block_Man project folder on the book's web page, www.sybex.com/go/introducingmaya2015, for a good reference of the grouping structure. This file shows you what a skeleton hierarchy does.

In the Hypergraph Hierarchy window, choose Options → Layout → Freeform Layout to position the nodes any way you want. To make selections easier, you can arrange the nodes as if they were on a body (see Figure 9.6). You can toggle between freeform and automatic, and your freeform layout will be retained.





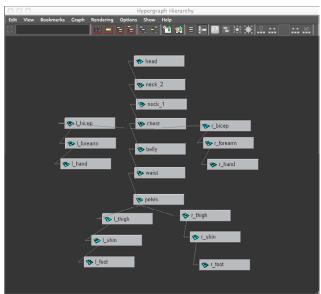


Figure 9.6

A freeform layout in the Hypergraph Hierarchy window

Creating the Skeleton

The basis of how the Block Man is laid out and grouped is what skeletons are all about. Skeletons make character animation easier by automating, at the very least, the hierarchy and pivot placement described earlier.

THE PELVIS AS ROOT

Traditionally, the pelvis is the basis of all biped setups. The root of any skeletal system (whether using bones or geometry as the example) is the character's pivot point—the center of balance. Because a biped character centers itself on two feet, its pelvis becomes the root of its skeletal system. In CG, the pelvis becomes the parent node of the whole system and is the node used to move or orient the entire character. In a skeleton system, this would be the root joint.

The root is then the top parent of the system below it and runs the entire chain. Therefore, selecting character parts straight from the Outliner or the Hypergraph is sometimes easier. You can see that a good naming convention is always important with character setups.

You'll use the Block Man to create a simple skeleton. Load block_man_v01.ma from the Block_Man project. This is the same as block_man_v02.ma, but this version isn't grouped.

1. Maximize the front view window. Switch to the Animation menu set by using the drop-down menu or by pressing F2.

- 2. Activate the Joint tool by choosing Skeleton → Joint Tool. Your cursor turns into a cross.
- 3. Click in the middle of the pelvis to place the first joint, which is the root joint of the skeleton.
- 4. Shift+click up to the space between the pelvis and the waist.

The joint display sizes in your Maya window may not match those shown in the book. This isn't a problem, however; you can change the joint sizes by clicking Display → Animation → Joint Size.

By pressing Shift as you click, you create a joint in a straight line from the last joint placement. A bone is created between the two joints as a visual guide to the skeleton. The placement of the joints depends on the active view.

5. Click more joints up the spine at the gaps between the body parts, as shown in Figure 9.7. Place the second-to-last joint at the base of the skull, as shown in the side view in Figure 9.7. Place the last joint at the top of the head. Then press Enter to exit the Joint tool. Select the top joint at the head and, in a side view, move that joint toward the forehead as shown to make the angle in Figure 9.7.

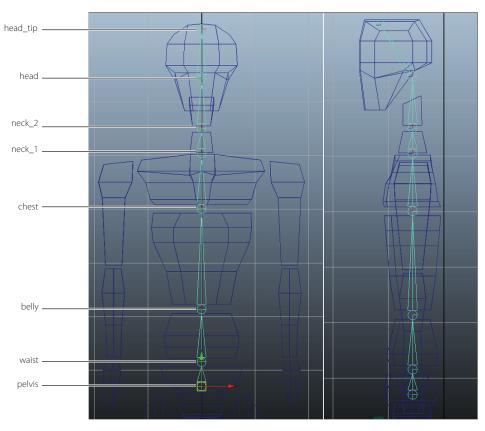


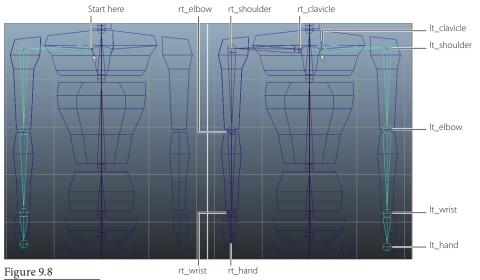
Figure 9.7 Place spine joints straight up the middle of the body and offset the last joint

in the head.

6. Use Figure 9.7 as a guide to name all your joints. Now you need to start a new branch of joints leading into the legs and arms. Begin with the arms.

Pressing the Up arrow key takes you up one node in a hierarchy (a.k.a. pick-walking). Pressing the Down arrow key takes you down one node in a hierarchy. This approach also applies to skeletons because they're hierarchies.

- 7. Enter the Joint tool (Skeleton → Joint Tool). In the front view, click to place a new joint at the top of the screen-left side of the upper chest just to the side of the joints you've already placed (Figure 9.8, left); then click to place a second connected joint in the bicep at the shoulder. Click down to create joints at the elbow, the wrist, and the tip of the character's right hand (which is on the left side of your screen). Press Enter to complete that part of the skeleton.
- 8. Reenter the Joint tool and repeat the previous step to create joints for the other arm, as shown in Figure 9.8 (right). Name your joints!
- 9. You'll notice that the arms are not part of the main skeleton yet. You need to connect them to the main skeleton. Open the Outliner and select the lt_clavicle joint. MMB+drag it to the chest node to group it to the chest (Figure 9.9). Notice that a new bone appears between these two joints. Group the rt_clavicle to the chest as well. Figure 9.10 shows the arms are now part of the main skeleton.



Place joints for the arm.

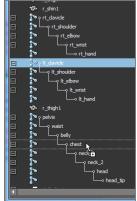
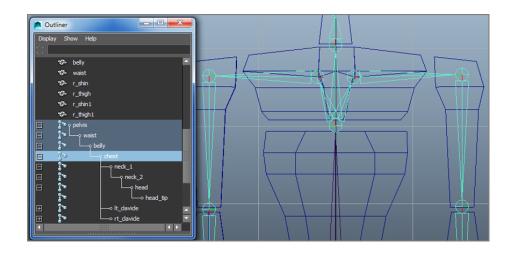


Figure 9.9 Group the It clavicle to the chest.

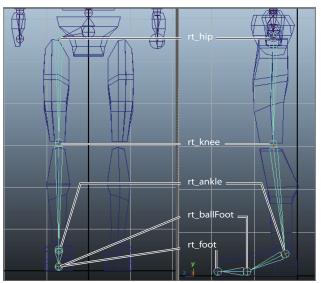
Figure 9.10

The arms are grouped to the chest, and the joints are connected with new bones.



- 10. To start another string of joints in the first leg, enter the Joint tool. Then click the pelvis joint, selecting it. In the front view, place the first leg joint at the top of the screen-left thigh (the character's right thigh). Place the next joint between the thigh and knee and then another at the top of the foot where it meets the ankle. Do not exit the Joint tool yet!
- 11. Switch to the side view and place another joint at the ankle, another in the middle bottom of the foot, and the last joint at the tip of the foot. Press Enter to exit the Joint tool, and you'll see that your leg is already attached to the pelvis; there's no need to group in the Outliner this time like with the clavicles.

Figure 9.11
Place the leg joints and name them.



- 12. In the side view, select the knee joint, and enter Pivot mode by pressing and holding the D key. Very slightly move the knee joint away from the front of the character, out toward the toe. This slight bend will be needed later when you continue the character rig. Using Pivot mode, make sure your joints are placed properly for the leg and foot using Figure 9.11 as a guide. Name your joints!
- 13. Select the rt_hip joint and press Ctrl+D to duplicate the hip and the joints beneath it. In the front view, move the new rt_hip1 joint over to the other leg and place it as shown in Figure 9.12 (left). Rename those new joints as shown.

You're all done! You'll be adding to this later in the chapter as you create a more wellrounded rig for the character, but for now, check your work against Figure 9.12 (right) and the block_man_skeleton_v01.ma file from the Block Man project.

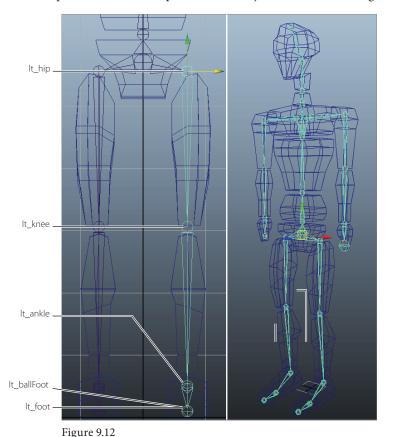
Attaching to the Skeleton

You now have a full skeleton for your character. To attach the geometry, all you need to do is parent the body parts under their appropriate joints. Before you get to that, take a few minutes and make sure all your joints are named in the Outliner to make the scene easier to manage. You can use Figure 9.13 as a naming reference.

You can also load the block_man_skeleton_v01.ma file from the Block_Man project to get to this point.

To parent the Block Man's geometry to the skeleton, follow these steps:

1. Starting with the pelvis, parent it under the pelvis joint by MMB+clicking and dragging it to the pelvis joint. Once it's in the group, MMB+drag the pelvis_geom again and place it between the pelvis and waist joints, as shown in Figure 9.13.



Duplicate the leg joints, move them to the other leg, and name the new joints (left). The skeleton is complete (right).

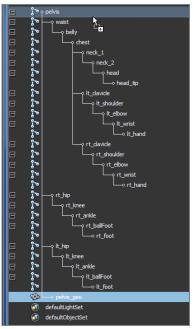
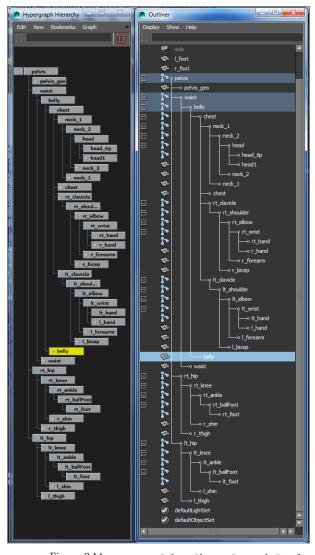


Figure 9.13 After you check your naming, you'll parent the pelvis geometry under the pelvis joint.



- 2. Continue the MMB+dragging process in the Outliner to place all but the feet geometry in its proper place in the skeleton using the following list and Figure 9.14 as guides. You will deal with the feet a little differently. Figure 9.14 shows the finished hierarchy in the Outliner and Hypergraph.
 - Shins under the knees; thighs under the hips
 - Hands under the wrists; forearms under the elbows
 - Biceps under the shoulder joints
 - Head under the head joint (not the head_tip joint)
 - Top neck geometry (neck_2) under neck_2 joint
 - Bottom neck geometry (neck 1) under the neck_1 joint
 - Chest under the chest joint
 - Belly under the belly joint
 - Waist under the waist joint
- 3. Now you'll bind the feet to the foot joints, instead of grouping them. Since there is a joint in the middle of the foot, you'll want to bend the foot. You will explore binding in more detail later in the chapter, but for now, select the l_foot geometry in the Outliner; then press Ctrl and select all three foot joints (lt_ankle, lt_ballFoot, and lt_foot) in the Outliner as well (Figure 9.15, left).

Figure 9.14

Parent all the geometry except the feet into its proper place in the skeleton. The finished hierarchy is shown in the Hypergraph and Outliner side-by-side.

- 4. Select Skin → Smooth Bind → Options, set Bind To to Selected Joints and Bind Method to Heat Map, and click the Bind Skin button. This will bind (a.k.a. skin) the foot geometry to the joints so you can then bend, or deform, the foot.
- 5. To test the bind, select the middle foot joint (lt_ballFoot) and rotate it; the tip of the foot should move. Select the ankle and rotate it, and the whole foot should move. Undo your rotations or set the rotations back to 0 so that the foot is at its rest pose (a.k.a. bind pose) again. Just don't undo the binding unless you need to redo steps 3 and 4. (See Figure 9.15, right.)
- 6. Repeat steps 3 and 4 for the other foot to bind the geometry and test its movement. Just remember to undo your rotations to return to the bind pose.

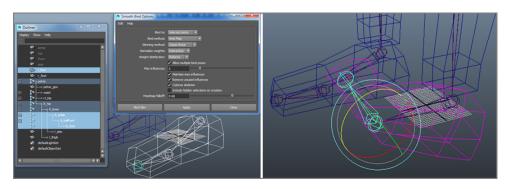


Figure 9.15

The Smooth Bind Skin Options: the left foot geometry and the left foot joints selected (left). Bind and test the left foot (right).

The Block Man is now set up with a skeleton that you'll use to make a simple walk cycle. This exercise will help get you more familiar and comfortable with animating a character. You will do more to this setup to make a better animation rig for the Block Man character later in the chapter.

The Block Man: FK Walk Cycle

A walk cycle is an animation that takes the character through a few steps that can be repeated many times so that the character seems to be taking numerous steps. In a cycle, make sure the position of the first frame matches the position of the last frame so that when the animation sequence is cycled, no "pop" occurs in the motion at that point.

Now, try animating this character's walk cycle using FK on the skeleton. You'll find the workflow straightforward, as if you were adjusting positions on a doll.

Load the block_man_skeleton_v02.ma file from the Block_Man project from the companion web page for the properly grouped and bound model and skeleton. In this file, the right arm and leg have drawing overrides enabled: This makes their wireframe display white to make it easier to distinguish between the right and left sides while working.

Use the key poses in the following figures to guide you in animating the body as it walks. You'll key at five-frame intervals to lay down the gross animation. You can go back and adjust the timing of the joint rotations in the Graph Editor to make the animation better later. The white leg and arm are behind the body, farther from the camera.

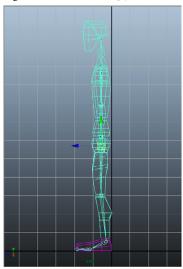
This animation is also called *pose animation* because you're posing the character from keyframe to keyframe.

Starting Out: Frames 1 and 5

Figure 9.16 shows the character's starting position. Here, you'll set a key for this position and then begin the walk cycle by moving the joints into their second position and keyframing that.

1. At frame 1, set a key for the rotation of all the joints. The easiest way is to select all the joints (and only the joints) in the Outliner or the

Figure 9.16 The character's starting position



- Hypergraph. With pose animation, you have to make sure that all the joints are keyframed at every step, even if Auto Keyframe is turned on. Also set a position keyframe for just the pelvis joint.
- 2. Go to frame 5. Rotate the back leg back (the Block Man's right, white leg), rotate the rt_ballFoot joint to bend the foot, and bend the ankle to make the ball of the foot level again. Lower the body (select and move the pelvis joint) to line up the back heel with the ground. This will keep the man on the ground as he goes through the walk cycle, although he won't actually move forward yet.
- 3. Rotate the near leg (the man's left blue wireframe leg) forward, bend the knee, and pivot the foot up a bit.

SELECTING ONLY JOINTS IN THE OUTLINER

A quick way to select all the joints, and only the joints, is to filter the Outliner view to show only joints. In the Outliner, choose Show → Objects → Joints. To reset the Outliner, choose Show → Show All.

- 4. Rotate the back white wireframe arm forward and rotate the near arm back (opposite from the legs). Bend the arms at the elbows.
- 5. Bend the man forward at the waist, bend neck_1 forward, and tilt the head back up to compensate a little. Figure 9.17 shows the pose at this point.
- 6. Select all the joints in the Outliner and set a rotation key. You're setting a pose for all the joints, which will ensure that all the body parts are in sync.

If you don't key everything every step of the way, some parts of the body won't key with Auto Keyframe properly because the last time they moved may have been two steps previous. This may cause confusion, so it's best to key every joint at every step until you feel more comfortable editing character animation.

Frame 10

Figure 9.18 shows the position you'll keyframe at frame 10; it's approximately midstride for the first leg.

- 1. Go to frame 10. Rotate the back leg out further and level the foot. Lower the body to place the man on the ground.
- 2. Rotate the front leg out, straighten the knee, and flatten the foot to place it on the ground. This is midstride. Swing the arms in their current direction a touch more. Bend the torso forward some more. Make sure you set a key for all the joints.

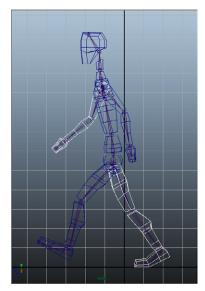


Figure 9.17 The second pose (frame 5)

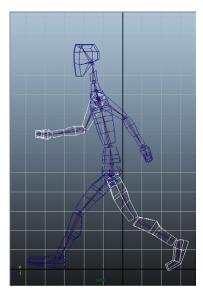


Figure 9.18 The third pose (frame 10)

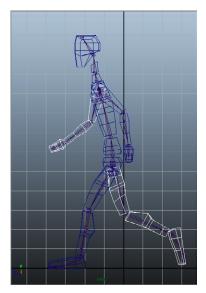


Figure 9.19 The fourth pose (frame 15)

Frame 15

Figure 9.19 shows the position you'll keyframe at frame 15. At this point, the character begins to shift his weight to the front leg as it plants on the ground, and the character also begins lifting the back leg.

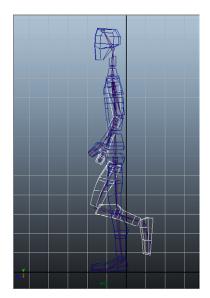
- 1. Go to frame 15. Rotate the front leg back toward the body and raise the body as the man steps to keep the front foot flat on the ground. Rotate the back knee up to lift the foot and rotate the foot down to make him push off the toe.
- 2. Start swinging the arms in the opposite direction. Start straightening the torso back up, but bend the head forward a bit.

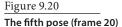
Frame 20

At frame 20, the man will shift all his weight onto the front leg and move his body over that leg, lifting his rear leg to begin its swing out front to finish the stride. Figure 9.20 shows the pose.

Follow these steps:

- 1. Rotate the front leg almost straight under the man and lift up the body to keep the front foot on the ground. Lift the rear leg and swing it forward.
- 2. Straighten the torso and keep the arms swinging in their new direction. Key all the joints.





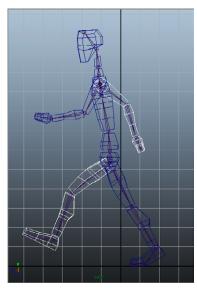


Figure 9.21 The sixth pose (frame 25)

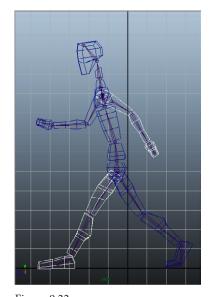


Figure 9.22 The seventh pose (frame 30)

Frame 25

Now, the man will swing his whole body forward, pivoting on the left leg (the dark one) to put himself off center and ready to fall forward into the next step. Figure 9.21 shows the pose. Here are the steps:

- 1. Go to frame 25.
- 2. Rotate the front (dark) leg back behind the man and swing the white leg up and ready to take the next step. Lower the body to keep the now rear foot (the dark one) on the ground.

Frame 30

Use Figure 9.22 as a guide for creating the next pose. Notice that it's similar to the pose at frame 10. As a matter of fact, the only major differences are which leg and arm are in front. Everything else should be about the same. You'll want some variety in the exact positions to make the animation more interesting, but the poses are very similar.

Completing the First Steps

You've finished a set of poses for the character's first step. You animated the left leg taking a step forward in the first series of poses. The next series of poses has to do with the right leg. The pose at frame 35 corresponds to the pose at frame 15. Frame 40 matches frame 20. You will copy keyframes in the Graph Editor to make more steps, but first let's retime these initial keys before continuing.

When a 30-frame section is complete, you need to return to the animation through the Graph Editor. Adjust all the keyframes that you initially set at these five-frame intervals

Logistically speaking, some poses take a little less time to achieve than the evenly spaced five frames you used. For example, achieving the second pose from the start position should take four frames. The third pose (see Figure 9.18, earlier in the chapter) from frame 5 to frame 10 should take four frames. The next frame section, originally from frame 10 to 15 (the fourth pose; see Figure 9.19, earlier in the chapter), should take only three frames. To accomplish this easily, follow these steps:

- 1. Select the top node of the skeleton (the pelvis) and open the Graph Editor. On the pelvis node in the left side of the Graph Editor, Shift+click the plus sign to open the entire tree of nodes beneath the pelvis. All the animated channels show their curves, as shown in Figure 9.23 (top).
- 2. Marquee-select all the keyframes on the curves beyond frame 1, not including those at frame 1. Press the W key to activate the Translate tool. Shift+MMB+click in the Graph Editor (so you can move in only one axis) and drag the keys horizontally to move them all 1 unit (frame) to the left. All the keyframes move, and the second pose now goes from frame 5 to frame 4.

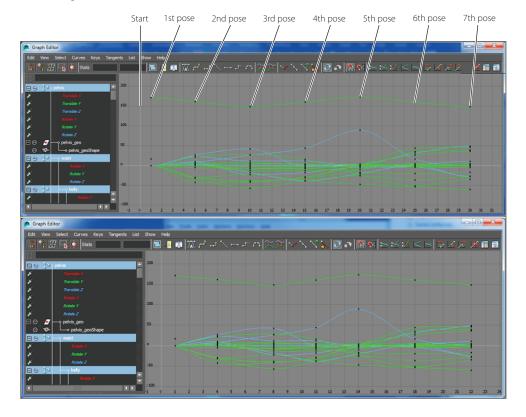


Figure 9.23

The Graph Editor shows the walk animation curves (top). The curves are retimed (bottom).

- 3. Deselect the keys now at frame 4 by holding down the Ctrl key (you also use the Ctrl key on a Mac) and marquee-deselecting all those keys at frame 4. Shift+MMB+click and drag the remaining selected keys 1 unit to the left again. The third pose goes from being at frame 9 to frame 8.
- 4. Deselect the keys now at frame 8 and Shift+click and drag the other selected keys to the left two frames so that the fourth pose animates between frame 8 and frame 11. Deselect the frame 11 keys and move the rest over two frames to the left again so that the next section runs from frame 11 to frame 14. The following section should go from frame 14 to frame 18. The final section should go from frame 18 to frame 22. Figure 9.23 (bottom) shows the new layout for the curves.
- 5. Continue to set and adjust keys for another cycle or two of the walk. The majority of time spent in animating something like this involves using the Graph Editor to time out the keyframes to make the animation look believable. Also, try offsetting some of the arm rotations a frame to the left or right to break up the monotony that arises from having everything keyed on the same frame.

Load the file block_walk_v01.mov or block_walk_v01.avi of this walk cycle from the Images folder of the Block_Man project on the companion web page to see the animation in motion. It's a rough cycle, and you have to keep adjusting the character's height to keep the feet on the ground. This is where IK comes in handy, as you'll see later in this chapter. Also, the file block_man_skeleton_walk_v01.ma in the Block_Man project has the keyframed steps for you to play with and continue animating.

Copy and Paste Keyframes

Now that you have a base to work with, you'll copy and paste keyframes to extend the animation. This can get a little tricky, so take this slow if you're not comfortable with the Graph Editor yet. Open the file block_man_skeleton_walk_v01.ma to follow along with these steps:

- 1. Extend the Time Range slider to 18 frames and open the Graph Editor. Select the back (white) leg at the rt_hip joint.
- 2. In the Graph Editor, Shift+click the plus sign next to the rt_hip entry to see all the joints and all their curves beneath it in the hierarchy.
- 3. Marquee-select all the keys from frame 4 to frame 22. In the Graph Editor window, select Edit → Copy.
- 4. Move the Time slider to frame 30 and, in the Graph Editor window, select Edit → Paste \square . In the option box, set the Paste method to Merge. This setting is very important; otherwise, the curves will not paste properly for your use! Click Apply to paste the keys at frame 30. Figure 9.24 shows the pasted frames in the Graph Editor and the Paste options.

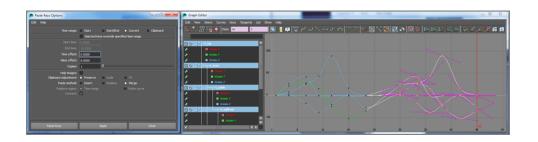


Figure 9.24
Copy and paste keys for the white colored leg.

- 5. Select the other leg (lt_hip) and repeat steps 2 through 4 to copy all the keys from frames 4 through 22. Go to frame 30 and paste (with the Merge option as in step 3) the keys for this leg. Don't forget to Shift+click to expand all the joints beneath the hip joint in the Graph Editor to see, copy, and paste all the joints' keys as you did in step 1.
- 6. Repeat steps 2 through 4 for each of the two arms (rt_shoulder and lt_shoulder) to copy and paste their keys as well (copying keys from frame 4 through 22 and pasting at frame 30). Again, don't forget to Shift+click them in the Graph Editor (step 2)!
- 7. If you scrub your animation in the Timeline all the way to frame 48, you can verify that the animation copied properly for the arms and legs.
- 8. The body is not moving yet, so let's copy and paste those keys now. In the Outliner, Ctrl+select the waist, belly, chest, neck_1, neck_2, and head joints.
- 9. Repeat steps 3 and 4 to copy and paste keys from frames 4 through 22 to frame 30 as you did with the legs and arms. There is no need to Shift+click any of the joints in the Graph Editor (step 2) this time since you individually selected each of the joints you need. Figure 9.25 shows the pasted frames for the body movement.

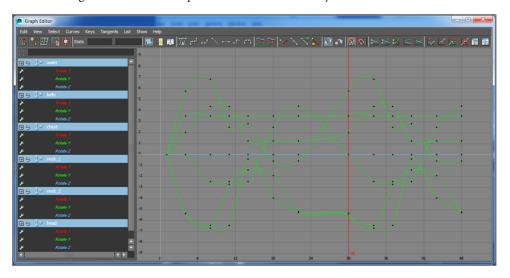


Figure 9.25
Copy and paste keys for the body

movement.

- 10. Scrub your animation, and you'll see you need to copy and paste keys for the pelvis next. Select the pelvis joint.
- 11. In the Graph Editor, select all the keys from frame 1 to frame 22. This time, go to frame 27 and paste (with the Merge option as before!) the keys.
- 12. Scrub your animation, and everything should be working great! But around frame 25, you should see the character dip down too far. Simply raise him up and key the pelvis position at frame 25 to compensate, and you're done! The Block Man is walking for a full 48-frame clip now.

Open the file block_man_skeleton_walk_v02.ma from the Block_Man project to check your work to this point.

Walk Cycle Wrap-Up

This walk cycle animation is more about getting comfortable with keyframing and skeletons than it is about creating great walk animation, so take some time to practice and get better. Character animation takes a lot of time and patience, and I encourage you to keep tweaking this animation and even create different walks of your own. Animating walk cycles is a good way to hone your skills. Several great books are devoted to character rigging and animation alone, and you can research the field for ways to become more proficient. But keep in mind that movement and timing are what make animation good, not the setup or the model.

Skeletons: The Hand

For another foray into a skeletal system, you can give yourself a hand—literally. You'll use a skeleton to deform the geometry and animate it as a hand would move.

Load the file poly_hand_skeleton_v01.ma from the Poly Hand Anim project from the companion web page. The hand is shown in Figure 9.26 (top).

You'll use it to create a bone structure to make the hand animate. This process is called rigging.

Rigging the Hand

To create the first bones of the hand, follow these steps:

- 1. Maximize the top view window. Switch to the Animation menu set by using the drop-down menu or by pressing F2.
- 2. Activate the Joint tool by choosing Skeleton → Joint Tool. Your cursor turns into a cross.
- 3. Click at the base of the wrist to place the first joint. This will be the root joint of the hand.
- 4. Shift+click the bottom part of the palm.
- 5. Place joints down through the thumb from this second joint, according to the corresponding bones in Figure 9.26 (bottom).

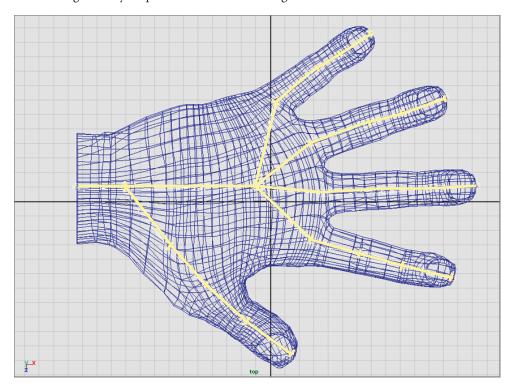
Figure 9.26

The hand mesh (top); placing joints in the hand (bottom)

- 6. To start another string of joints into the palm, press the Up arrow key four times until you're at the second joint at the base of the palm.
- 7. The next joint you place will be a branch from this joint. Place that joint in the middle of the palm. Place another joint up further along the palm and then branch it out to the index finger. Press the Up arrow key to return to that upper palm joint and start a new branch into the middle finger.

Repeat this procedure to place joints for the remaining fingers, as shown in Figure 9.27. With these joints placed, you have a simple skeleton rig for the hand. This rig allows you quite a bit of hand and finger movement.

Figure 9.27 The joints in the hand



Check the other views (see Figure 9.28) to see where you need to tweak your joint positions to fit the hand. Ideally, you want the joints to be set inside your intended geometry in the same way that real bones are laid out.

To position the joints, you can use either of two Maya tools: Move or Move Pivot. First you'll try the Move tool. Select the tip joint for the pinkie. It needs to be lowered into the pinkie itself. Select the Move tool (press W), and move it down into the tip of the pinkie. Now, move on to the top pinkie knuckle. Notice that if you move the knuckle, the tip moves as well. That's not such a great idea.

Instead, it's best to move joints as pivots. Because joints are nothing more than pivots, go into Move Pivot mode (hold down the D key to activate Move Pivot, or press the Ins key on Windows or the fn+Home key on a Mac) to move joints. Select the top pinkieknuckle joint, and move it with Move Pivot instead. Only the joint moves, and the bones adjust to the new position. Set the positions on the remaining joints to be inside the hand properly, as shown in Figure 9.29 and Figure 9.30.

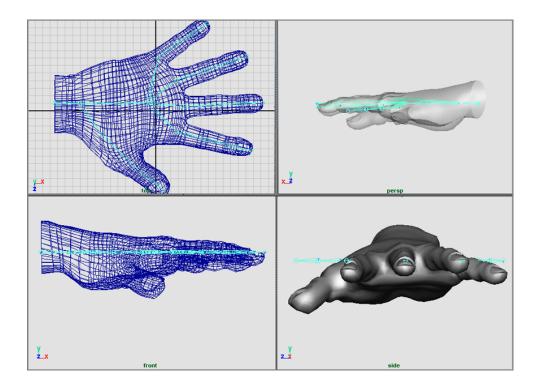


Figure 9.28
Four views of the hand with initial placement of the joints

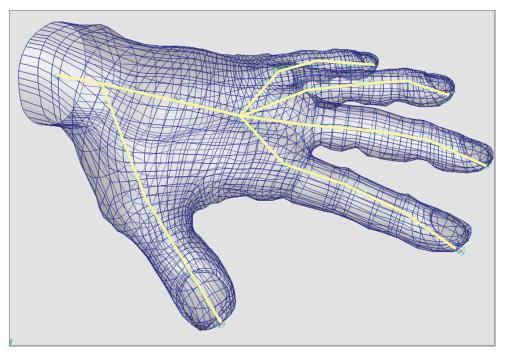
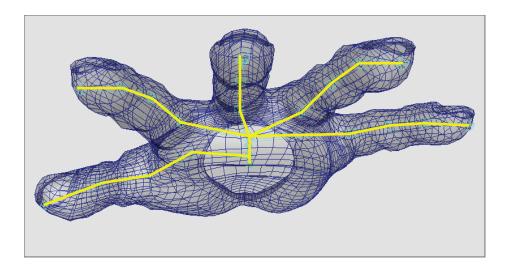


Figure 9.29
The joints of the hand placed properly in the geometry



Binding to Geometry

An integral part of rigging a character or an object with a skeleton is *binding*, also known as *skinning*. Binding is another way to attach geometry to a skeletal system. With the Block Man, you directly attached the whole pieces of geometry to the bones through parenting, whereas binding involves attaching *clusters*, or groups of vertices or CVs, of the geometry to the skeleton to allow the skeleton to deform the model. This is typically how skeletons are used in character animation work. (For more on grouping and parenting, refer to the solar system exercise in Chapter 2, "Jumping into Basic Animation Headfirst.")

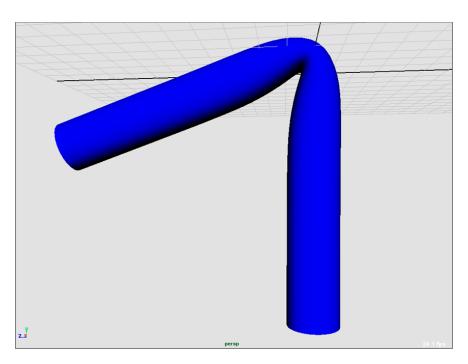
The basic technique of binding a character is easy. However, Maya gives you tremendous control over how your geometry deforms.

Binding Overview

Binding is, in theory, identical to the Lattice deformer you saw in Chapter 5, "Modeling with NURBS Surfaces and Deformers." A lattice attached to an object exerts influence over parts of the model according to the sections of the lattice. Each section affects the CVs of a NURBS surface or the vertices of a polygon surface within its borders—and as a section of the lattice moves, it takes those points of the model with it.

Skeletal binding does much the same thing. It attaches the model's points to the bones, and as the bones pivot around their joints, the section of the model that is attached follows.

By attaching vertices or CVs (depending on your geometry) to a skeleton, you can bend or distort the geometry. When a bone moves or rotates about its joint, it pulls with it the points that are attached to it. The geometry then deforms to fit the new configuration of the bones bound to it.

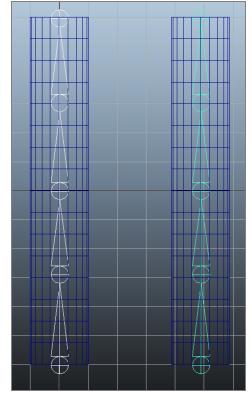


You can bind geometry to a skeleton in two ways: using Smooth Bind or Interactive Bind. Figure 9.31 shows a cylinder with a smooth bind. An interactive skin bind will yield similar results to the smooth bind. The only real difference between the two methods is the editing capabilities once you have bound the skin. We'll take a look at these differences later in the chapter.

With a Smooth Bind, we have two common binding methods that we'll compare here, using Closest Distance and Heat Map. Create a polygon cylinder, with Height Divisions of 16. The more spans you have in the deformable model, the better it will bend. Scale the cylinder in Y to a scale of 5 to make it tall. Duplicate the cylinder, and move it over in your window in the X-Axis. Now, in the front view, create a four-bone (five-joint) skeleton that starts at the bottom of the first cylinder and goes straight up the middle, ending at the tip. Duplicate the skeleton, and move it to the center of the second cylinder (Figure 9.32).

Figure 9.32

Make two cylinders
with skeletons
in them.



CREATING A SMOOTH BIND WITH CLOSEST DISTANCE

To create a smooth bind, select the root of the first skeleton and its cylinder and choose Skin → Smooth Bind □.

In the option box, you'll find the Bind To parameter at the top set to the default of Joint Hierarchy (set it to Joint Hierarchy if it is different). You'll also find, under the Bind Method drop-down menu, the options Closest Distance, Closest In Hierarchy, Heat Map, and Geodesic Voxel. Closest Distance disregards a joint's position in the hierarchy of the skeleton and assigns influences according to how far the point is from the joint. The Heat Map method uses a "heat diffusion" technique to create influence falloff radiating out from the joint, and is arguably the most commonly used for character work as it assigns skin weights best suited for character movement, giving you creases at bending areas that are more believable than with the Closest Distance method.

Max Influences sets a limit on how many joints can affect a single point. Dropoff Rate determines how a joint's influence diminishes on points farther from it. For example, with Smooth Bind, one shoulder joint can influence, to varying degrees, points stretching down the arm and into the chest and belly. By limiting these two parameters, you can control how much of your model is pulled along by a particular joint.

Set Bind Method to Heat Map and click Bind Skin in the option box to smooth-bind your first cylinder to the bones using Closest Distance (the default).

CREATING A SMOOTH BIND WITH HEAT MAP

Now select the root of the second skeleton and the second cylinder and choose Skin → Smooth Bind □. This time, set the Bind Method to Heat Map.

Bend both cylinders to get a feel for how each creases at the bending joints. Figure 9.33 shows the difference. The Heat Map binding on the right shows more defined creases at the joints, much like fingers, while the Closest Distance binding on the left creates a smoother arc over the entire cylinder.

Interactive Skin Bind is practically the same as Smooth Bind and yields similar results to the smooth binding using Closest Distance shown in Figure 9.33, but the editing of the influences is slightly different than for Smooth Bind. You will explore Interactive Skin Bind with the hand model later in this chapter.

Figure 9.33

Closest Distance and Heat Map-Smooth-bound cylinders. The cylinder on the left is **Closest Distance** bound, and the one on the right is Heat Map bound.

DETACHING A SKELETON

If you want to do away with your binding, select the skeleton and its geometry, and choose Skin → Detach Skin. The model will snap to the shape it had before the bind was applied and the joints were rotated. It's common to bind and detach skeletons several times on the same model as you try to figure out the exact configuration that works best for you and your animation.

If you need to go back to the initial position of the skeleton at the point of binding it to the model, you can automatically set the skeleton back to the bind pose after any rotations have been applied to any of its joints. Simply select the skeleton and choose Skin → Go To Bind Pose to snap the skeleton and model into the position they were when you bound them together. It's also best to set your skeleton to the bind pose whenever you edit your binding weights.

Binding the Hand: Closest Distance Method

Try skinning the hand with a Smooth Bind.

Load your own hand or the poly_hand_skeleton_v02.ma file from the Poly_Hand_Anim project from the web page. You can also access a video tutorial to help you with this exercise from the book's website. Now, follow these steps to smooth-bind the hand:

Select the skeleton's root at the wrist, and Shift+click the hand as shown in Figure 9.34.
 Choose Skin → Smooth Bind □. Make sure the Bind Method is set to Closest Distance and click Bind Skin.

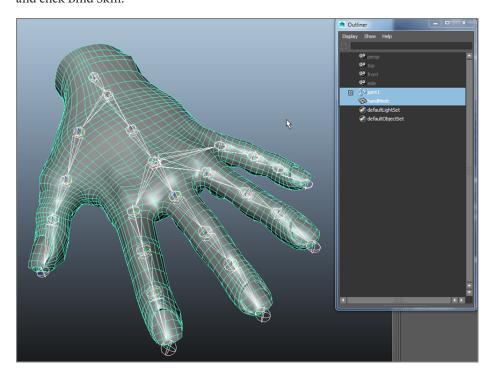


Figure 9.34
Select the root joint as well as the top node of the hand.

- 2. Try rotating some of the knuckle joints to see how the fingers respond. Go back to the bind pose when you're finished.
- 3. Rotate the middle knuckle of the index finger down. Notice how the knuckle gets thinner the more you bend the finger there. Go to the top knuckle of the index finger and rotate that joint. Notice that part of the hand moves with the finger. Figure 9.35 shows the result of bending at the index finger.

Figure 9.35 Bending at the index finger causes some unwanted deformation.

Editing a Smooth Bind

You usually edit a smooth bind by *painting skin weights*. Because points on the model are influenced by multiple joints in a smooth bind, you need to adjust just how much influence is exerted by these joints on the same points.

1. Make sure you're in Shaded mode (press 5). Select the hand and then choose Skin → Paint Skin Weights Tool □.

You paint skin weights on the affected geometry and not on the joints themselves, so you need to select the model and not the skeleton before invoking this tool.

- 2. Your hand should turn black, with a bit of light gray at the wrist (or perhaps at the index finger if that joint is selected). The option box appears, listing the joints that are connected to the hand, as shown in Figure 9.36.
- 3. The color value (between white and black) determines how much binding influence the selected joint in the option box is exerting on that part of the geometry. It's best to name your joints properly so that selecting from this window is easier and more intuitive. If you loaded the file from the web page, you need to name the joints yourself to organize the scene and make working with it easier.



Figure 9.36 The option box for the Paint Skin Weights tool

4. In the option box, select joint9, the index finger's top knuckle, and make sure the Paint Operation button under the Influence section is set to Replace. Change the Value slider to 0. In the Stroke section, Radius(U) and Radius(L) govern the size of your brush. In the Influence section, make sure the Opacity slider is set to 1.

To change the size of your Artisan Brush while you're painting weights, you can hold down the B key and drag the mouse left or right to adjust the radius of the brush interactively.

5. Click and paint a black color around parts of the hand and palm that shouldn't be affected by the index finger bending at its top knuckle (joint9), as shown in Figure 9.37.

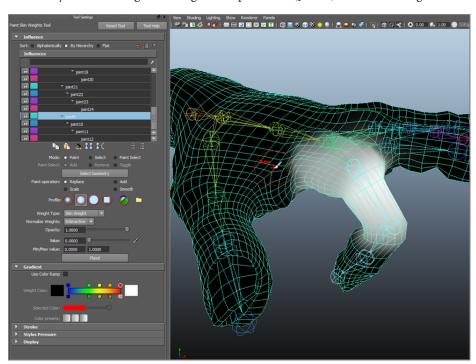


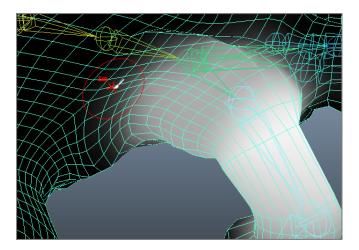
Figure 9.37

Paint the new weights to avoid unwanted deformations in the hand.

6. Smooth out the area where it goes from white to black. In the option box, in the Influence section, set Paint Operation to Smooth. Right-click to smooth the area around the knuckle for a cleaner deformation, as shown in Figure 9.38. Your index knuckle should now bend beautifully.

Skin weights must always be normalized in a smooth bind, meaning the values have to add up to 1. When you reduce the influence of a joint on an area of the surface, the influence amount is automatically shifted to other joints in the hierarchy that have influence over that area; those joints are now more responsible for its movement.

Figure 9.38 Smoothing out the bend at the index finger.



You can exit the Paint Skin Weights tool by selecting another tool (press W for Translate, for example), and your view will return to regular Shaded mode. Try bending the rest of the fingers and painting their influences; then, animate the hand, making gestures or grabbing an object using FK animation to set keys on the rotations.

When you paint weights on polygons, keep in mind that you're painting using the UVs. You may need to re-create the UVs of a polygonal mesh with a UV projection map for the Paint Weights tool to function properly, especially when you're importing and exporting the weight maps from one mesh to another (a procedure you won't encounter until later in your Maya experience).

The scene poly_hand_skeleton_v03.ma from the Poly Hand Anim project from the companion web page has the hand smooth-bound with painted weights on just the index finger for your reference. Try painting the other knuckles as needed for your animation.

Rigging work is essential for getting a good animation from your model. In a professional shop, it usually falls under the domain of a technical director (TD) who oversees the setup of characters and may also model their geometry. The more time I spent rigging scenes for the animators when I was a TD on the television show South Park, the easier and faster they were able to accomplish their animations.

Binding the Hand: Heat Map Method

Now let's try the same exercise using a Heat Map binding method instead of Closest Distance. Load your own hand model before it was bound or the poly_hand_skeleton_v02.ma file from the Poly_Hand_Anim project from the web page. Now, follow these steps to smoothbind the hand with the Heat Map method to compare to the previous method we tried:

1. Select the skeleton's root at the wrist and the hand mesh and choose Skin → Smooth Bind □. Change the Bind Method to Heat Map and click Bind Skin.

2. Rotate the middle knuckle of the index finger down. Go to the top knuckle of the index finger and rotate that joint as well. Notice that the bending is cleaner than before. Not much of the hand moves when you bend the index finger joints as it did in the previous exercise using Closest Method. Figure 9.39 shows the result of bending at the index finger having used the Heat Map method of the Smooth Bind.

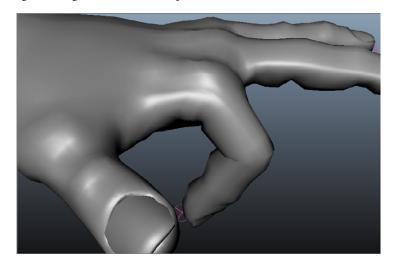


Figure 9.39
The index finger seems to bend more cleanly than before.

- 3. Select the hand mesh and choose Skin → Paint Skin Weights Tool □. Just as before, the hand turns black except for white areas on the mesh corresponding to the selected joint (shown in Figure 9.40 with the index finger's joint9 selected).
- 4. Go ahead and paint weights on this hand, and you'll notice that it is a little easier now that we've used Heat Map instead of Closest Distance.

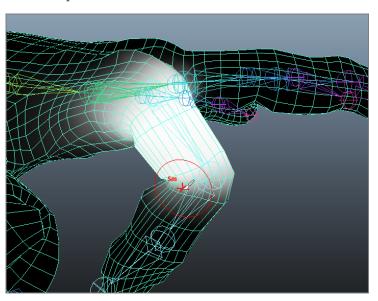


Figure 9.40

Painting skin weights on the Heat Map method bound hand.

The Heat Map method gives you cleaner creases and bends at the knuckles, where Closest Point was a bit more soft, often deforming parts of the hand that shouldn't be affected much by a bending finger. There's still some painting to be done to make the fingers bend perfectly even when using the Heat Map method. The poly_hand_skeleton_v04.ma file from the Poly_Hand_Anim project has the hand with the index finger with painted weights as a reference for you.

Binding the Hand: Interactive Bind

Now, try skinning the hand with Interactive Bind.

Load your own hand or the poly_hand_skeleton_v02.ma file from the Poly_Hand_Anim project from the web page.

The Interactive Bind method is a bit easier to control when compared to the painting of the Smooth Bind weights.

- 1. Select the root joint at the wrist and the handMesh object and choose Skin → Interactive Bind. Figure 9.41 shows the color scheme of a Volume manipulator that allows you to set the initial skin weights of the hand easily.
- 2. In the Outliner, select the different joints of the hand to see how the interactive binding shows the volume influences. Then select the top knuckle of the index finger (joint9). Figure 9.42 shows the influence. Select the red circle at the end of the Volume manipulator and size it down to reduce the influence on the middle finger, as shown in Figure 9.43.

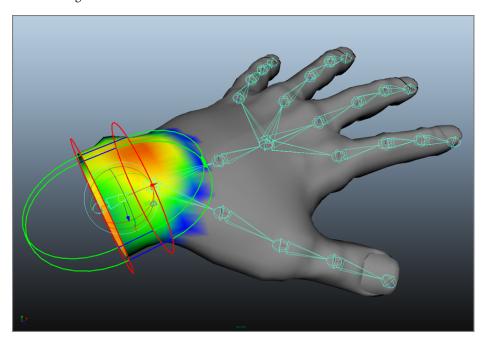


Figure 9.41

The Volume manipulator shows the influence of the selected joint at the wrist.

Figure 9.42
The index finger's first knuckle influences the middle finger, too.

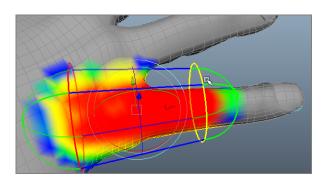


Figure 9.43
Use the Volume manipulator to reduce the influence on the middle finger.

3. Grab the green spherical end, and you can see that you can move that in or out to lengthen or shorten the Volume manipulator. You can use the move or rotate handles to move the volume as well. Make adjustments to the finger to remove influences on the middle finger and other undesired areas of the hand, as shown in Figure 9.44.

Editing the interactive skin bind, as you can see, is easy using the Volume manipulators. If you exited the tool when you first created the bind and no longer see the Volume manipulators, you can access them by choosing Skin -> Interactive Skin Bind Tool. When you access this tool, the Volume manipulators appear again, and you're able to adjust your skin's influences.

When you are happy with the proper level of influence, keep picking the other joints in the index finger to make sure your binding is proper. Then continue to the other fingers

to set up the binding on the hand properly. This is easier than painting weights as before, though not as controllable. Test the bind by bending the joints. This is just a more interactive way to bind a mesh than Smooth Bind, and may be more or less effective depending on the model you are binding. It does, however, use the Closest Distance method by default, and Heat Map method is not available for the Interactive Bind. You can, however, paint skin weights when you use Interactive Bind, just as you did with the Smooth Bind.

In the case of this hand and skeleton setup, the most effective binding has been using a Smooth Bind with the Heat Map method, with some subsequent skin weight painting.

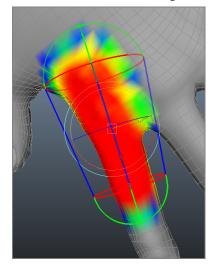


Figure 9.44
Set the influences by adjusting the Volume manipulator.

Inverse Kinematics

With IK, you have tools that let you plant a foot where it needs to be so you're not always moving the skeleton or model to compensate and keep the heel in place.

For legs, IK is nothing short of a blessing. There is no clearly preferable workflow to suggest when dealing with rigging arms and hands, however. Many people use IK on hands as well, but it can be better to animate the legs with IK and animate every other part of the body with FK. IK is best used when parts of the body (such as the feet) need to be planted at times. Planting the hands isn't necessary for a walk cycle, and having IK handles on the arms may create additional work while you are animating them. You will create a more well-rounded character rig that uses IK and easy-to-use character controls at the end of the chapter. First, let's get familiar with how IK works.

Rigging IK Legs

Back to the Block Man. Switch to that project and load your version or the block_man_ skeleton_v02.ma file from the Block_Man project from the companion web page.

You'll create an IK chain from the hip to the ankle on each foot. Creating the IK from the hip to the toe won't work.

Because IK automatically bends the joints in its chain according to where its end effector, or IK handle, is located, it has to choose which way to bend at a particular joint. This is why you created the legs slightly bent in the Block Man rig earlier in the chapter.

If you did not do this or your setup has straight legs for whatever reason, select the two knee joints and, in Pivot mode (hold down the D key), move the knees forward a bit over

> the feet to create a slight crook in the legs. Don't go too far; a slight amount is enough. This lets the IK solver know which way those joints are supposed to go.

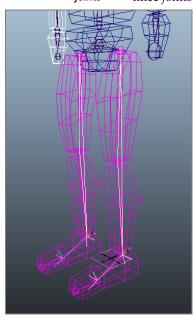
Now, on to creating the IK:

- 1. Open the IK Handle tool by choosing Skeleton → IK Handle Tool. Your cursor changes to a cross.
- 2. Select the start joint for the IK chain. This will be the root of this chain. Click the left thigh joint and then pick your end effector at the ankle joint. The bones in the IK chain turn brown. Repeat this procedure for the other leg. Figure 9.45 shows handles on both ankles.

If for some reason you can't manage to pick a joint for the IK tool, make sure Show → Pivots is turned on in your view panel. Also, if you have difficulty seeing the handles, you can increase their size by choosing Display → Animation → IK Handle Size.

3. Move the IK handles around to see how the legs react. When you're finished, reset the IK handle positions.

Figure 9.45 IK handles on both ankles with the roots at the hip joints



- 4. Grab the top joint of the skeleton, which is the pelvis joint. Move the joint, and the entire body moves with it. Select both ankle IK handles and set a translation key for them (press Shift+W). Grab the pelvis joint again and move it. The feet stick to their positions on the ground. Move the pelvis down, and the legs bend at the knees. Notice how the feet bend into the ground, though (see Figure 9.46, left).
- 5. Move the pelvis back to the origin. You can create an IK handle for the foot so that the foot stays flat on the ground. Open the IK Handle tool. For the start joint, select the first ankle; for the end effector, select the joint in the middle of that foot (lt_ball-Foot or rt_ballFoot). Repeat for the other foot.

You can invoke the last tool you used by pressing Y.

6. Set a translate key for the foot IK handles. Move the pelvis down; the legs bend at the knees and the ankle, keeping the feet flush on the ground (see Figure 9.49, right).

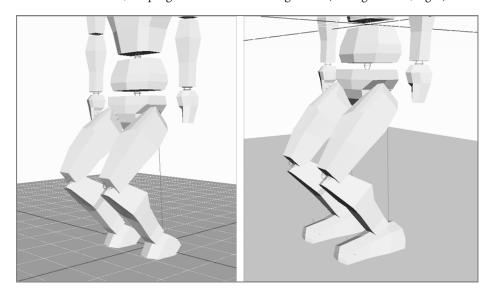


Figure 9.46

Creating another IK chain from the ankle to the tip of the foot and setting keyframes makes the feet stay on the ground (right) and not rotate into the ground (left).

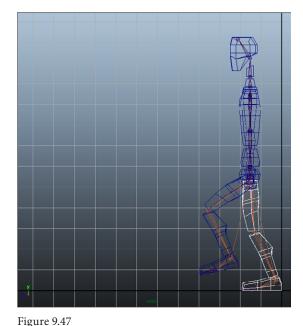
Creating an IK Walk Animation

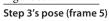
Because the Block Man's feet will stick to the ground, creating a walk cycle with IK animation is far easier than using FK (why didn't you tell us that before?). Making the animation look good is still a tough job that requires a lot of practice, though.

Load the scene file block_man_IK_v01.ma from the Block_Man project from the companion web page or use your own IK-rigged Block Man with handles at the ankles and feet. The white leg and arm are, again, on the far side of the character. You'll set keys every five frames again for the gross animation. To keep this short, I'll just discuss setting

poses with the feet. You can always return to the scene to add animation to the upper body with FK, as you did earlier in this chapter. Follow these steps:

- 1. On frame 1, set translate keys on the pelvis joint and all four IK handles for their start position.
- 2. Go to frame 5 and move the pelvis forward about 1 unit. The legs and feet lift off the ground a bit and strain to keep their position, but they stay back. Lower the pelvis to get the feet flat on the ground again. Set a key for the pelvis. Because Auto Keyframe is turned on, all keys are set for this animation. (With the FK animation, you set keys for everything at every pose.)
- 3. Grab both near IK handles for the ankle and foot (blue leg) and move them forward and up to match the pose shown in Figure 9.47.
- 4. Go to frame 10. Move the front foot forward and plant it on the ground. Move the pelvis another three-fourths of a unit. Set translation keys for the rear ankle and foot handles where they are. Be sure to place the pelvis so that the rear foot is almost flat on the ground. Match the pose shown in Figure 9.48.
- 5. Go to frame 15. Move the pelvis another 2 units to center the body over the front foot. Lift the rear ankle and foot IK handles up to bend the knee and bring the knee up a bit. Match the pose shown in Figure 9.49.
- 6. Go to frame 20. Move the pelvis forward 1 unit and swing the white leg forward as in the pose shown in Figure 9.50.





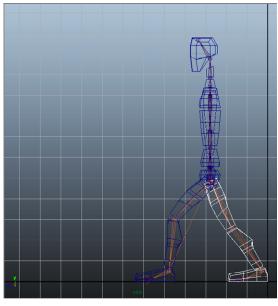
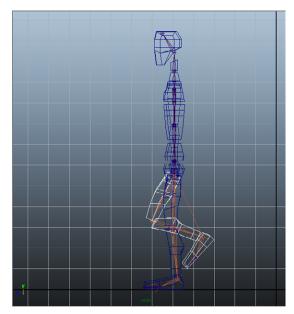


Figure 9.48

Step 4's pose (frame 10)



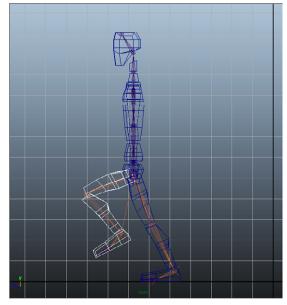


Figure 9.49
Step 5's pose (frame 15)

Figure 9.50
Step 6's pose (frame 20)

7. Move the pelvis three-fourths of a unit forward and plant the front leg down. Set keys for the rear leg and foot where they stand. Match the pose shown in Figure 9.51.

The next pose should match the pose in frame 10, although with the other leg. Continue the cycle, with each successive pose matching the one 15 frames before it on the opposite side.

At the end of this chapter, you'll take this one step further and create a simple character animation rig for the entire Block Man so you can have a nicely functioning character for animation.

Further Uses for IK Chains

Many animators use IK chains more often in effects animation than in character work. IK chains can drive whips and ropes, flutter flags, bounce ponytails, and pump pistons as well as move legs and arms. For example, you can use a different type of IK chain, the *spline* IK chain, to control the shape of your bone chain with a NURBS spline. This IK chain is great for snakes and other long, deforming objects.

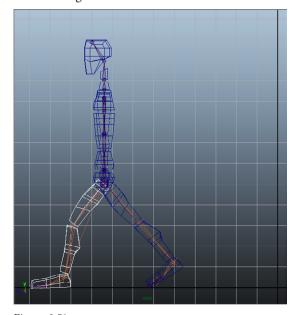
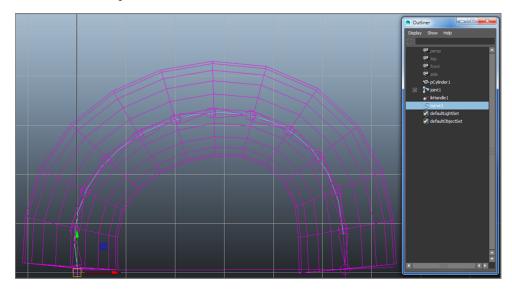


Figure 9.51
Step 7's pose (frame 25)

To create a spline IK chain, choose Skeleton → IK Spline Handle Tool and then select your top joint and end effector. Maya creates a spline running the length of the bone chain. Adjusting the curvature of the spline in turn drives the bones, which in turn drive the geometry bound to them. Figure 9.52 shows a spline curve affecting the curvature of the bones in its spline IK chain.

Figure 9.52

A spline IK chain is driven by the curvature of a NURBS spline. Adjusting the curve's CVs moves the joints.



Basic Relationships: Constraints

As you know, Maya is all about the relationships between object nodes. You can create animation on one object based on the animation of another object by setting up a relationship between the objects. The simplest way to do that (outside of grouping) is to create a constraint. For example, you can "glue" one object to another's position or rotation through a constraint.

A constraint creates a direct relationship between the source and the target object's Translate, Rotate, or Scale attributes. This section explores six types of constraints: point, orient, scale, aim, geometry, and normal.

The Point Constraint

To attach a source object to a target object but have the source follow only the position of the target, use a *point constraint*. A point constraint connects only the Translate attributes of the source to the target. To use this method, select the target object(s) and then Shift+click the source object. In the Animation menu set, choose Constrain → Point □.

Constraints are based on the pivots of the objects, so a point constraint snaps the source at its pivot point to the pivot point of the target and keeps it there, even in animation. But the options allow you to set an offset that creates a gap between the source and the target.

You can constrain the same source to more than one target object. The source then takes up the average position between the multiple targets. By setting the Weight slider in the option box, you can create more of an influence on the source by any of the targets.

In Figure 9.53, a cone has been point-constrained to a sphere. Wherever the sphere goes, the cone follows. This is different from parenting the cone to the sphere in that only its translations are affected by the sphere. If you rotate or scale the sphere, the cone won't rotate or scale with it.

Although you can blend keyframe animation with constraint animation, as a beginner to Maya, consider that after you set a point constraint like that shown in Figure 9.53, you're unable to control the cone's Translate attributes because they're being driven by the sphere's translations.

Point constraints are perfect to animate a character carrying a cane or a sword, for example. The rotations on the sword are still free to animate, but the sword is attached to the character's belt and follows the character throughout the scene.

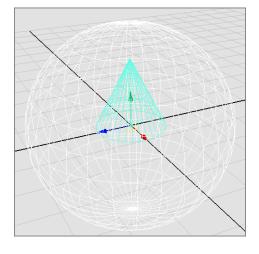


Figure 9.53 A cone point that is constrained to a sphere follows that sphere's position.

The Orient Constraint

An orient constraint attaches the source's Rotation attributes to the target's Rotation attributes. Select the target object(s) first and then Shift+click the source object. In the Animation menu set, choose Constrain → Orient □.

The Offset parameter allows you to set an offset in any axis. Otherwise, the source assumes the exact orientation of the target. In the case of multiple targets, the source uses the average of their orientations. Figure 9.54 shows the cone's orientation following an elongated sphere (the target).

A rotation constraint saves a lot of hassle when you have to animate an object to keep rotating in the same direction as another object. For example, you can use the rotation of one wheel of a locomotive to drive the rotation of all the other wheels.

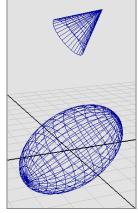


Figure 9.54 The cone's rotations match the sphere's rotations.

The Point on Poly Constraint

A point on poly constraint attaches a source object to a vertex of a mesh. Select the target object's vertex first and then Shift+click the object you want to place at that point (Figure 9.55, left). In the Animation menu set, choose Constrain \rightarrow Point On Poly \square .

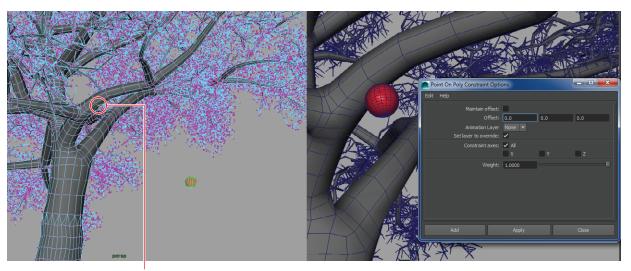


Figure 9.55

Selected vertex

The red ball is placed on the tree branch like a fruit with a point on poly constraint.

The object is snapped to the vertex of the target at its pivot point. Even if the target object is animated and deforming, like a character, the object will stay on that vertex. Figure 9.55 (right) shows the sphere pinned to the branch at the selected vertex location.

The point on poly constraint is good for pinning objects together, such as leaves on a branch.

The Aim Constraint

The aim constraint adjusts the source's rotations so that the source always points to the target object. Select the target object(s) first and then Shift+click the source object. In the Animation menu set, choose Constrain → Aim □.

The aim constraint has more options than the other constraints because you need to specify which axis of the source is to point to the target. You do so using the Aim Vector and Up Vector settings.

The Aim Vector setting specifies which axis of the source is the "front" and points to the target. In the cone and sphere examples, you set the Aim Vector option of the cone to (0,1,0) to make the *Y*-axis the front so that the cone's point aims at the sphere. If Aim Vector is set to (1,0,0), for example, the cone's side points to the sphere. Figure 9.56 shows the cone pointing to the sphere with an Aim Vector setting of (0,1,0).

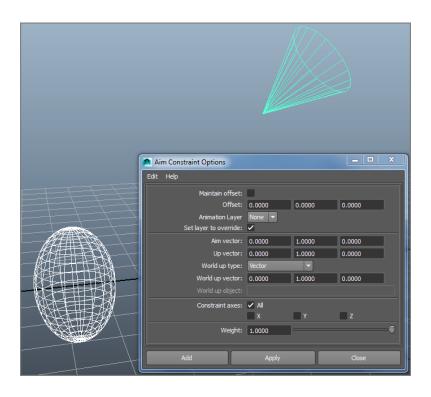


Figure 9.56 The cone aiming at

the sphere

The Offset values create an offset on the source's Rotation attributes, tilting it one way or another. The Up Vector setting specifies which way the cone faces when it's pointing to the sphere.

Aim constraints are perfect for animating cameras to follow a subject, such as a car at a racetrack.

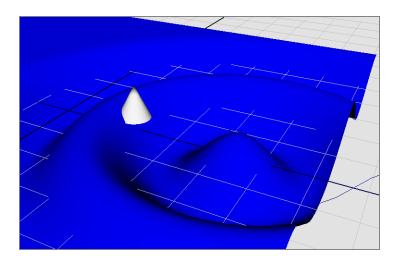
Geometry and Normal Constraints

The geometry and normal constraints constrain the source object to the surface of the target object (as long as it's a NURBS or poly mesh).

With a geometry constraint, the source object attaches, at its pivot point, to the surface of the target. It tries to keep its own position as best it can, shifting as its target surface changes beneath it. Again, select the target, select the source object, and choose Constrain → Geometry.

A geometry constraint is useful when you want to keep an object on a deforming surface, such as a floating boat on a lake. Figure 9.57 shows the cone after it has been geometryconstrained to a NURBS plane that is being deformed by a Wave deformer (choose Create Deformers → Nonlinear → Wave). The cone sits on the surface as the waves ripple through, but it doesn't rock back and forth to stay oriented with the surface.

Figure 9.57
With a geometry constraint, the cone sits on the deforming surface.



To get the cone to orient itself so that it truly floats on the surface, you need to use a normal constraint. Using a normal constraint rotates the cone to follow the surface's normals, keeping it perpendicular to the surface.

A *surface normal* is an imaginary perpendicular tangent line that emanates from all surfaces to give the surface direction.

The normal constraint is similar to the aim constraint, and its options are similar. Using the Aim Vector setting, you specify which way is up for the object to define the orientation that the source should maintain. However, this setting doesn't constrain the location of the source to the target. If you want a floating effect, use geometry and a normal constraint to get the cone to bob up and down and roll back and forth as the waves ripple along (see Figure 9.58).

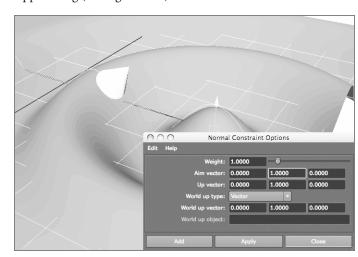


Figure 9.58

The cone now animates to float on the water surface, using both geometry and normal constraints.

Scale, Parent, Tangent, and Pole Vector Constraints

Four more constraints are possible in Maya: the scale, parent, tangent, and pole vector constraints. Simply, a scale constraint attaches the source's Scale attributes to the target's Scale attributes. A parent constraint constrains an object's translation and rotation to another object by mimicking a parent-child relationship without actually parenting the objects. This keeps objects aligned without worrying about any grouping issues. You'll have a firsthand look at this in the exercise where you rig the locomotive later in this chapter. Lucky you!

A tangent constraint keeps an object's orientation so that the object always points along a curve's direction. This constraint is usually used with a geometry constraint or path animation to keep the object traveling along a curve pointed in the right direction, no matter the direction of the curve. A point on poly constraint allows you to select a vertex on a poly mesh and constrain an object to that vertex. Pole vector constraints are used extensively in character animation rigs to keep IK joints from flipping beyond 180 degrees of motion.

Basic Relationships: Set-Driven Keys

A favorite feature for animation riggers is the set-driven key (SDK). An SDK establishes a relationship for objects that lets you create controls that drive certain features of a character or an object in a scene.

Before you can use an SDK, you must create extra attributes and attach them to a character's top node. These new attributes drive part of the character's animation. The term character is used broadly here. For example, you can set up a vehicle so that an SDK turns its wheels.

Let's start with a simple SDK relationship between two objects. You'll create a relationship between a ball and a cone. As the ball moves up in the Y-axis, the cone spins in the X-axis. As the ball descends, the cone spins back. You'll then revisit the hand and set up an SDK on the skeleton that animates the model.

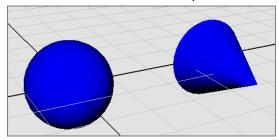
Creating a Set-Driven Key

To create a simple SDK to make a sphere control the animation of a cone's rotation, follow these steps:

1. Create a NURBS sphere and a cone in a new scene. Move the cone to the side of the sphere and lay it on its side, as shown in Figure 9.59.

2. Select the sphere and, in the Animation menu set, choose Animate → Set Driven Key → Set. The Set Driven Key window opens with the nurbsSphere1 object selected in the lower half of the window (the Driven section). Its attributes are listed on the right, as you can see in Figure 9.60.

Figure 9.59 Lay out a cone and a sphere.





The Set Driven Key window

- 3. You want the sphere to drive the animation of the cone, so you need to switch the sphere to be the driver and not what's driven. Click the Load Driver button to list the sphere in the top half of the window.
- 4. Select the cone and click the Load Driven button to display the cone's attributes in the bottom half of the window.
- 5. In the Driver section, select the sphere's Translate Y attribute. In the Driven section, select the cone's Rotate X attribute. Click the Key button to set an SDK that essentially says that when the sphere is on the ground (Y = 0), the cone's X rotation is 0 because both attributes are currently 0. The cone's Rotate X attribute turns orange in the Channel Box, meaning a driven key has been set.
- 6. Select the sphere and raise it in *Y* to a height of 5. Select the cone and rotate it in *X* to **1800** to make it spin properly. Click the Key button in the Set Driven Key window to specify that when the sphere is at a height of 5, the cone's Rotate X attribute is 1800 degrees. As the sphere's height increases from 0 to 5, the cone spins from 0 to 1800 in X.

An Advanced Set-Driven Key: The Hand

Automating some animations on a character is indispensable to an animator. This can't be truer than when setting up an SDK for hand control. After you model and bind a hand to a skeleton, you're ready for an SDK.

Open the scene poly_hand_skeleton_v05.ma from the Poly Hand Anim project from the companion web page or use your own file that has the hand and its skeleton and is bound to the skin. Your file shouldn't have animation, though. Set your hand to the bind pose before you begin.

Creating a New Attribute

First you'll create a new attribute called index pull to control a contracting finger.

- 1. Select the hand. In the handMesh tab of the Attribute Editor, click the Extra Attributes section. For now, at least, this section is empty.
- 2. In the Attribute Editor menu, choose Attributes → Add Attributes to open the Add Attribute window, which is shown in Figure 9.61. In the Long Name field, enter **index_pull**. Maya will automatically display that attribute as Index Pull in the UI. Make sure the Make Attribute Keyable option is selected and that the Float option is selected in the Data Type section. In the Numeric Attribute Properties section, set Minimum to **0**, Maximum to **10**, and Default to **0**. Click OK.

After you click OK, the Index_Pull slider appears in the Attribute Editor and the Channel Box when you select the handMesh object. This attribute alone will control the entire index finger.

Assigning the Set-Driven Key

To set up the relationships with the SDK, follow these steps:

- 1. With the handMesh selected, open the Set Driven Key window (choose Animate → Set Driven Key → Set...). Click Load Driver to specify that the hand should drive the animation.
- 2. Because you're animating the index finger pulling back, you want to drive the rotations of the top three knuckles. Deselect the hand, and then shift+click all three knuckles on the index finger to select them. Click the Load Driven button. All three knuckles appear on the bottom.
- 3. Select the hand's Index Pull attribute and the three knuckles' Rotate Y attributes, as shown in Figure 9.62.
- 4. With the rotations of the knuckles at 0 and the index_pull attribute at 0 as well, click the Key button to set the first relationship. When index_pull is at 0, the finger is extended.
- 5. Select the handMesh node, and set the index_pull attribute to 5.

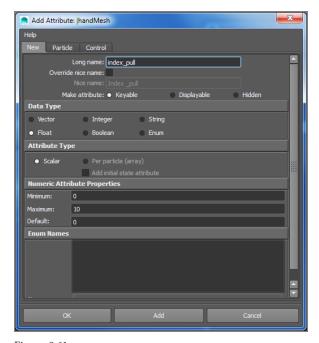


Figure 9.61

The Add Attribute window



Figure 9.62

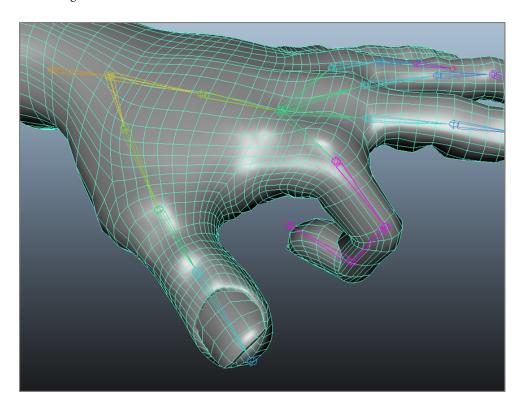
The Set Driven Key window for the hand

- 6. Select the fingertip's knuckle (joint11 in the web page file) and rotate it in Y to 20. Select the next joint up the chain (the middle knuckle, joint10) and rotate it to 35 in the Y-axis. Select the final index knuckle (joint9) and rotate it in the Y-axis to 5.
- 7. In the Set Driven Key window, select the three joints' Rotate Y and the handMesh Index_Pull attribute and click the Key button. When the index_pull attribute is at 5, the finger assumes this bent position.
- 8. Select the handMesh, and set index_pull to 10.
- 9. Set the tip knuckle (joint11) to rotate to 65 in Y. Set the middle knuckle (joint10) to **60**. Set the last knuckle (joint9) to **50**. Click the Key button to see the result shown in Figure 9.63.

Select the handMesh node and change the value of the index_pull attribute to curl your index finger. All you need to do to animate the finger is to set keys on that one attribute! Furthermore, you can set up a single SDK to control the bending of all the fingers at once, or you can set up one SDK for each finger for more control.

Open the scene poly_hand_skeleton_v06.ma from the Poly_Hand_Anim project available on the companion web page to see the hand with the SDK set up on the index finger.





In this section, you'll use a locomotive to put your new animation skills to use. Download the file fancy_locomotive_anim_v1.mb from the Scenes folder of the Locomotive project from the book's web page; this scene is shown in Figure 9.64. Notice in this scene, there are small plus signs near the wheels. These are selection handles, allowing you to more easily select objects or groups of objects. They have been enabled for some of the objects on the locomotive to make it easier to work with. To toggle selection handles on or off for any node in Maya, Display \rightarrow Transform Display \rightarrow Selection Handles.

After selection handles for the locomotive's wheels and drive arms are turned on, only the objects that have selection handles will be selected when you make a marquee selection that covers the entire locomotive.

Setting Up Wheel Control

Your goal here is to rig the scene to animate all the secondary movements automatically based on some simple controls, such as you did for the hand earlier this chapter. In reality, the locomotive's steam pump drives the arms that then turn the wheels on the locomotive. You'll work backward, however, and use one wheel to drive the animation of everything else.

As a challenge, you can also try this exercise of rigging the locomotive wheel's to rotate in unison by using rotation constraints instead of the Connection Editor.

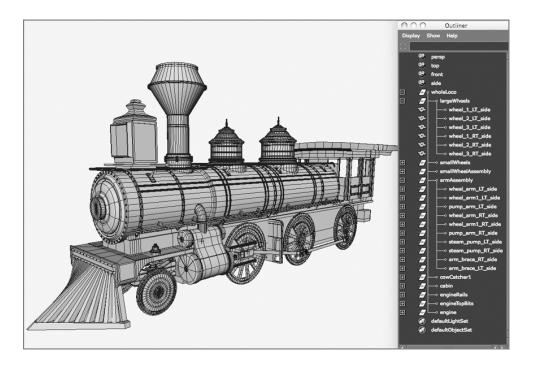


Figure 9.64
The fancier

The fancier locomotive model

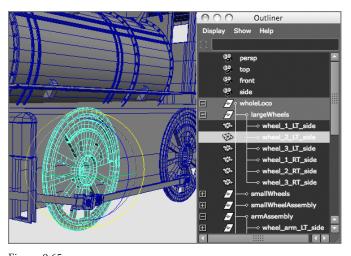


Figure 9.65 Select the middle wheel.



Figure 9.66 Connect the rotations of the two wheels.

Because all the large wheels have the same diameter, they rotate the same as the locomotive moves. In this case, you'll use the Connection Editor to attach the *X* Rotation on all the wheels to your main control wheel. You'll pick the middle wheel to be the control. To set up the locomotive, follow these steps:

- 1. Select the middle wheel on the left side of the locomotive (node wheel_2_LT_side), as shown in Figure 9.65. Open the Connection Editor (choose Window → General Editors → Connection Editor). Click the Reload Left button to load the attributes of the selected middle wheel. Now, select the front wheel on the left side, and click the Reload Right button.
- 2. Scroll down in the Connection Editor until you find Rotate in both columns. Click to highlight Rotate in the left column, and then click to highlight Rotate in the right column. Doing so connects the two rotations so that they both rotate at the same time, effectively letting you drive the animation of both wheels from just the center wheel. Figure 9.66 shows the Connection Editor.
- 3. Select the back wheel on the left side (wheel 3 LT side). Click the Reload Right button in the Connection Editor. Connect the Rotate attribute for the middle and back wheels. Close the Connection Editor, and select just the middle wheel. When you rotate the wheel, all three wheels rotate together.
- 4. Repeat this procedure to connect the rotations of the three wheels on the other side to this middle wheel as well. Now all six wheels rotate in sync with the one control wheel. When you select that left-side middle wheel (the control wheel), the other five wheels turn magenta, signifying a connection between these objects.

If you get strange results when you connect the rotations of objects (for example, if the wheels flip over or rotate the opposite direction of the control wheel), try disconnecting all the connections, freezing transforms, and reconnecting the attributes.

Controlling the Wheel Arms

You've now automated the animation of the wheels. Next, you'll figure out how to connect the wheel arms to the wheels and drive their motion as well. To do so, follow these steps:

- 1. Create a single joint that lines up with the first wheel arm. The root joint is placed where the wheel arm meets the middle wheel (control wheel), and the end joint is placed where the wheel arm meets the pump arm, as shown in Figure 9.67. The pump arm has been templated in this graphic (displays in light gray wireframe) to show you the entire wheel arm and joint.
- 2. Group the joint under the control wheel's node, as shown in the Outliner, earlier in Figure 9.65. Then, group the wheel arm under the top joint. This way, the joint rotates with the control wheel, also shown in Figure 9.68, albeit incorrectly for the pump arm.
- 3. As you saw in Figure 9.68, the joint isn't rotating properly to make the pump arm work right. The other end of it needs to attach to the pump arm in front of the front wheel, not fly up in space. You can use an IK handle for this. Make sure the rotation of the control wheel and the joint/wheel arm are set back to 0 to place them in the original position. In the Animation menu set, choose Skeleton → IK Handle Tool. Make sure the settings are reset for the tool. Select the root joint as the start joint for the IK handle. Select the other tip of the bone as the end effector. You now have an IK handle at the tip where the wheel arm connects to the pump arm, as shown in Figure 9.69.

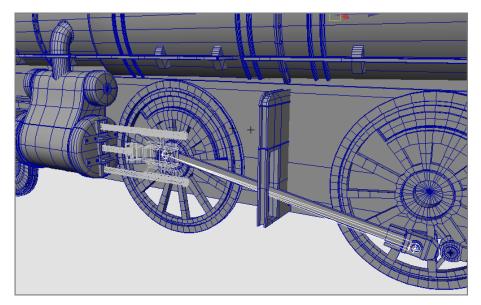


Figure 9.67

Create a joint from the middle wheel to the pump arm at the first wheel.

Figure 9.68

Group the top joint under the wheel and then group the wheel arm under the top joint.

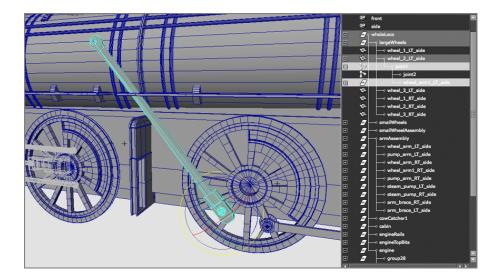
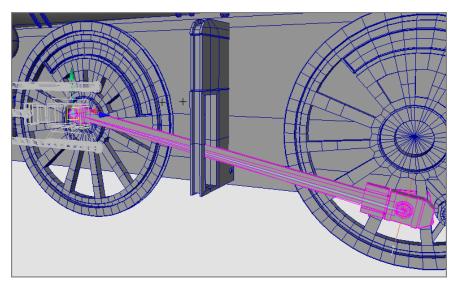


Figure 9.69

Place the end effector where the pump arm and the first wheel connect.

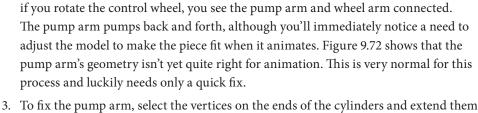


- 4. If you rotate the control wheel now, the wheel arm still separates from the pump arm. This is because the IK handle you just created needs a keyframe to keep it in position—that is, attached to the pump arm. Select the IK handle, and, at frame 1, set a position keyframe. Now, if you rotate the control wheel, the joint and wheel arm pump back and forth.
- 5. Group the IK handle (ikHandle1) under the top node of the locomotive (wholeLoco), as shown in Figure 9.70.

Controlling the Pump Arm

Next, you need to attach the pump arm to the wheel arm so that it pumps back and forth as the control wheel turns. If you simply group the pump arm with the end joint of the wheel arm's bone, the pump arm will float up and down as it pumps back and forth. You need to use a constraint to force the pump arm to move back and forth only in the Z-axis.

- 1. Make sure the control wheel is set back to 0 rotation. Select the pump arm, templated in Figure 9.71 so that you can see through to the wheel arm and joint, and line up its pivot with the end joint of the wheel arm bone.
- 2. Select the end joint (called joint2), Ctrl+click (or Cmd+click on a Mac) the pump arm group in the Outliner (called pump_arm_LT_side), and, in the Animation menu set, choose Constrain → Point □. In the option box, under Constraint Axes, select only Z to constrain the pump arm only in the Z-axis and click the Add button. Now if you rotate the control wheel, you see the pump arm and wheel arm connected. The pump arm pumps back and forth, although you'll immediately notice a need to adjust the model to make the piece fit when it animates. Figure 9.72 shows that the pump arm's geometry isn't yet quite right for animation. This is very normal for this process and luckily needs only a quick fix.



Now the pump arm won't pull out of the steam pump assembly. 4. Adjust the pump arm so that the geometry fits

to make them longer, as shown in Figure 9.73.

The scene file fancy_locomotive_anim_v2.mb will catch you up to this point. Compare it to your work.

when the pump pushes in as well.

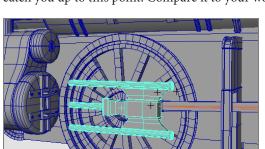


Figure 9.72 The pump arm is too short!

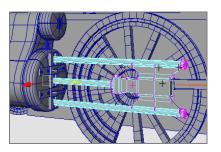
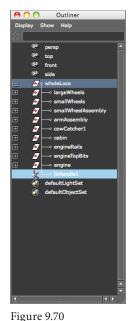


Figure 9.73 Use vertices to extend the pump arm.



Group the IK handle under the locomotive's top node.

Figure 9.71

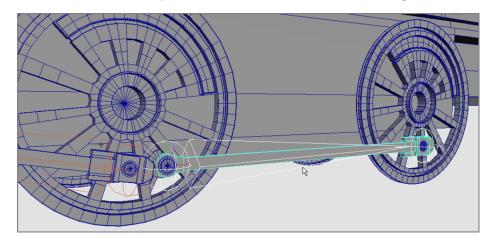
Line up the pivot of the pump arm with the end joint of the wheel arm joint.

Controlling the Back Wheel

All that remains is to control the animation of the back wheel and its wheel arm. To set up the wheel arm animation, follow these steps:

1. Using the methods described in the steps in the "Controlling the Wheel Arms" section, create a joint to follow along the wheel arm between the middle control wheel and the back wheel. The root of the joint is set at the control wheel, as shown in Figure 9.74.

Figure 9.74 Create a joint to control the back wheel arm.



2. As before, create an IK handle for the end joint of this new bone, where it meets the back wheel, as shown in Figure 9.75. Make sure the handle is at the back wheel, not the middle control wheel.

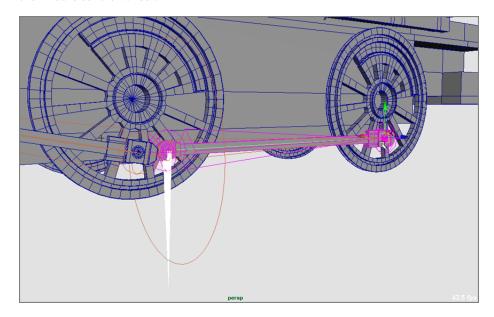


Figure 9.75 Create an IK handle to attach the wheel arm and the back wheel to the control wheel.

3. Group the new joint under the master wheel and then group the wheel arm under this new joint. If you rotate the control wheel, the wheel arm rotates with the joint and wheel but doesn't connect to the back wheel yet. You need to attach the IK handle you just created for that joint to the back wheel.

If you group the IK handle, as shown earlier in Figure 9.70, you'll run into a problem when you animate. Let's try it. Group the IK handle (ikHandle2) under the end back wheel, as shown in Figure 9.76, and then rotate the control wheel. The wheel arm pumps back and forth along with the back wheel, but every now and then the wheel arm geometry flips over backward. This isn't good.

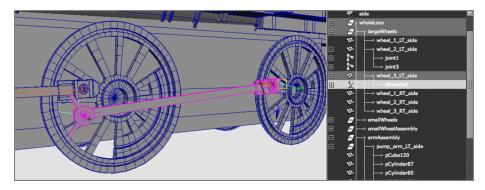


Figure 9.76

The wheel arm geometry flips over if you group the IK handle under the back wheel.

Fixing this is easy. The grouping of the IK handle to the back wheel is causing the issue. Although that is pretty much what you want to do, parenting the IK handle under the wheel is problematic. Here is where the parent constraint becomes extremely helpful. It gives you the desired result without the geometry flipping.

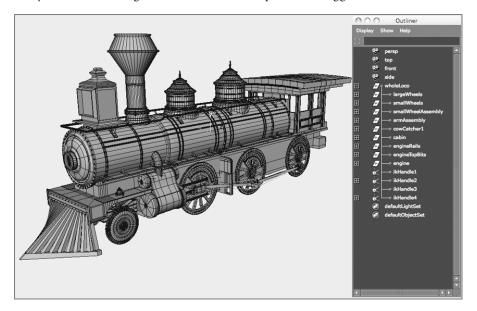
- 4. Make sure your control wheel is back to 0 rotation first. MMB+click in the Outliner and place the IK handle outside the hierarchy of the locomotive to remove the IK handle from under the back wheel's node. You may also undo your past actions to the point before you grouped the IK handle (ikHandle2) under the back wheel. (You have to love Undo!)
- 5. Select the back wheel, Shift+click the IK handle (ikHandle2) and choose Constrain → Parent. Now, if you rotate the control wheel, everything works great.
- 6. Group the IK handle (ikHandle2) under the top node of the locomotive (wholeLoco). You can use fancy_locomotive_anim_v3.mb to compare your work.

Again, seeing procedures go slightly awry, like when the wheel arm flipped over, is important. Doing so gives you a taste of trouble and a chance to fix it. Troubleshooting is an integral skill for a good CG artist.

Finishing the Rig

You're almost home free with the locomotive wheel rigging. Everything works great when you rotate the control wheel. If you select the top node of the locomotive and translate the train back and forth, everything should work perfectly. Repeat the steps in the previous few sections to connect the wheel arms and wheels on the other side of the locomotive, and you're finished! Figure 9.77 shows the completed and rigged locomotive.

Figure 9.77 The rigged fancy locomotive



Creating a Simple Character Rig

In this section, you will revisit the Block Man setup to create a more well-rounded character rig with controls like professional animators use. This rig was created by Maks Naporowski, a fellow instructor at USC and CG animator/rigger as a fairly simple biped rig for animation. Bear in mind that character rigging is an involved process, and you are starting to scratch the surface here. When you are done with this rig, you will have a simple two-legged character that you can easily animate using the controls you will set up based on what you've already accomplished throughout this chapter.

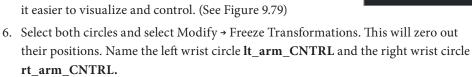
Creating Control Shapes

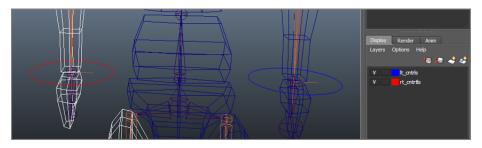
Animators hardly ever manipulate and keyframe IK handles or joints directly when a good rig is available to them, and that's what you should keep in mind for the following rig:

- 1. Open the scene file block_man_skeleton_v02.ma from the Block_Man project.
- 2. In the Animation menu set, choose Skeleton → IK Handle Tool □. In the option box, set the Current solver attribute to Rotate-Plane Solver. Click one of the hip joints and

then click the ankle joint. This makes an IK chain for the leg. Repeat for the other leg. Name them lt_ikHandle and rt ikHandle.

- 3. Create an IK chain for the arms, from the shoulder to the wrist, also making sure to use the Rotate-Plane Solver. Name them as well. Figure 9.78 shows all four IK handles created.
- 4. Create a circle (Create → NURBS Primitives → Circle). Scale it up and center it around the left wrist. Name this circle lt_arm_CNTRL. Duplicate the circle, move the copy to center on the right wrist, and name it rt_arm_CNTRL. Refer to Figure 9.79 for placement.
- 5. In the Channel Box/Layer Editor, click the Display tab in the Layer Editor and create two new layers: lt cntrls (make it blue) and **rt_cntrls** (make it red). Assign the left wrist circle to lt cntrls and the right wrist circle to rt cntrls. This makes it easier to visualize and control. (See Figure 9.79)





7. Create two more circles, and adjust their CVs to make them oval to fit around the feet. Size and place them around the feet as shown in Figure 9.80. Assign each foot oval to the appropriate lt_cntrls or rt_cntrls display layers and name the ovals lt_ foot_CNTRL and rt_foot_CNTRL, respectively.

8. Create a large circle and center it around the pelvis joint (name it **body_CNTRL**). Make another large square shape and place it on the floor around the feet (name it main_CNTRL). Assign these two shapes to a new display layer called cn_cntrls and make it green.

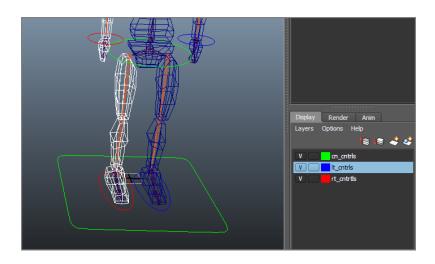
These shapes will be the primary controllers for the animation of the character. They are easy to select and manipulate and make animation much easier.

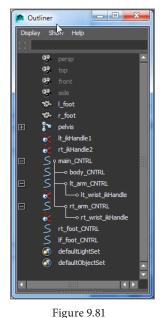
Figure 9.78 Create IK handles for the arms and legs.



Create circles for the wrists, place them, and assign them to their own display layers.

Create control shapes for the feet, pelvis, and body.





Grouping the controllers and IK handles

Setting Up the Controls

Now comes the tough part of rigging it all to work! You'll group the shapes and create relationships to the skeleton here:

- 1. Parent the body_CNTRL under the main_CNTRL shape.
- 2. Parent the two arm controllers (rt_arm_CNTRL and lt_arm_CNTRL) under the main_CNTRL shape as well.
- 3. Parent each of the two wrist IK handles under each respective circle shape, as shown in Figure 9.81.
- 4. Select the left wrist circle and the lt_wrist joint and choose Constrain → Orient □. In the option box, turn on Maintain Offset and click Add. This allows you to control the hand's rotation with the circle, as well as the hand's position. Select the wrist circle and move and rotate it around to see how the arm reacts. The elbow can swing around a little much, so you'll add a control for the twist of the arm.
- Select the left wrist circle and open the Attribute Editor. Choose Attributes →
 Add Attribute. Enter the name twist to the wrist circle and keep the options at
 their default (Data Type: Float).
- 6. Now you'll connect it to the IK handle. Select both the wrists' IK handle and the circle controller and choose Window → Node Editor. The white circles on either side of the nodes are input and output sockets. (See Figure 9.82.)

THE NODE EDITOR

The Node Editor is fairly new to Maya and gives you an easy way to create and manage connections between objects or nodes. You can navigate in it easily with Alt+mouse combinations like other Maya windows, and you can arrange the display of nodes however you want.



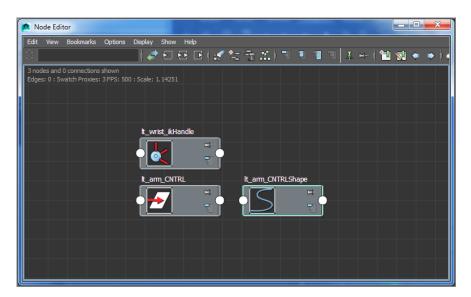


Figure 9.82

The Node Editor displays the IK handle and the circle control shape.

- 7. Click the output socket of the lt_arm_CNTRL node (the white circle on the right) and select Twist from the context menu. This connects a yellow rubber band to your cursor. Move the mouse to the input socket of the lt wrist ikHandle node (the white circle on the left) and click. Choose Twist from the menu. This automatically creates an intermediate node but essentially connects these two attributes. (See Figure 9.83.)
- 8. Now if you select the wrist circle and adjust the twist value, the arm will twist. If nothing happens, check your node connection and make sure you used the ikRPsolver when you created the IK handles in the previous section.
- 9. Repeat steps 4 through 8 for the right hand.

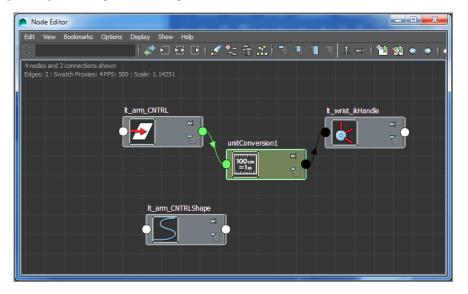


Figure 9.83

You've connected the Twist attributes.

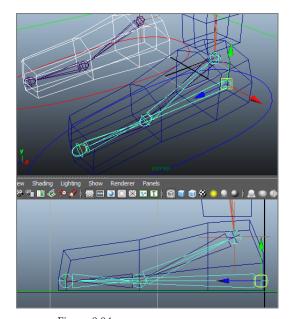


Figure 9.84 Create a reverse ioint chain for the left foot.

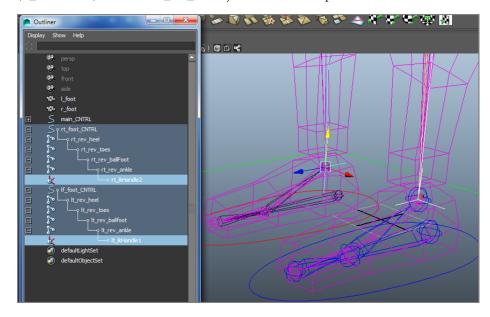
Figure 9.85 Parent the heel ioints and ankle IK handles.

- 10. Now let's do the same for the feet for the Twist attribute. Select the left foot's control oval and add a new attribute called twist. Select the oval and the ankle joint (lt_ ikHandlel) and open the Node Editor and connect the Twist attributes as you did in step 7 for the arm. Repeat for the right leg.
- 11. Select the pelvis joint and parent that under the body_ CNTRL circle.

Setting Up Heel Controls

Now let's move on to creating some nice foot controls to allow the character to stand on his tiptoes or his heels easily. For that you will need to build a reverse joint chain to control the foot from the heel to supplement the existing leg joints.

- 1. In the side view, choose Skeleton → Joint Tool and place a joint at the back of the left heel, as shown in Figure 9.84.
- Place the second joint exactly on the existing toe joint (lt_foot). Hold down V for point snaps when you create the joint so you can snap it exactly to that toe joint. Snap the third joint to the existing lt_ballFoot joint; finally, snap the fourth joint to the existing lt_ankle joint.
- 2. Grab the top joint of this reverse chain and parent it under the lt_foot_CNTRL oval. Name the joints as noted in Figure 9.85. Finally, parent the left ankle's IK handle (lt_ihHandle1) under the lt_rev_ankle joint as shown. Repeat for the other foot.



- 3. Next, parent both the foot control ovals under the main_CNTRL square. (See Figure 9.86.)
- 4. Select the main_CNTRL square and move the rig around. The character moves along with it. Select the body CNTRL circle. When you move the body control, the feet and hands stay because of IK (see Figure 9.87), and that's how you want it. Undo
- 5. Create a display layer called **rigging**. Select the lt rev_heel and rt_rev_heel joints you just created and parented and assign them to the new rigging display layer. This will make it easy to hide these extra reverse joints to keep them out of the way later.

your moves.

- 6. Select the lt rev toes joint first and then the lt ballfoot joint (see Figure 9.88 for the selections) and choose Constrain → Orient □. Make sure Maintain Offset is checked and click Apply. The constraint node should appear under the lt_ballFoot node, as shown in the Outliner in Figure 9.88.
- 7. Select the lt rev ballFoot joint and then the lt_ankle joint and create an Aim Constraint like in the previous step.



Figure 9.86 Parent the foot controls.

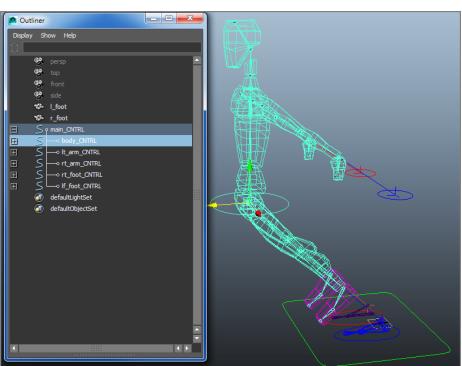


Figure 9.87

Moving the body control and the main control works perfectly.

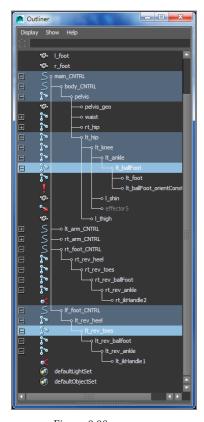


Figure 9.88 Aim constrain the two joints together.

- 8. Repeat steps 6 and 7 for the right foot.
- 9. When you're done, your Outliner should be similar to the one shown in Figure 9.88. If your constraint nodes are not in the right places, check to make sure you selected the nodes in order in Steps 6 and 7 when you made the Aim Constraint.

Set-Driven Keys: Heel Controls

Now let's set up some fancy heel controls.

1. Select the left foot control oval and add a new attribute to it called **heel**. In the options, change Data Type to Float, but give it a Minimum of -5 and a Maximum of 10 (see Figure 9.89). Repeat for the right foot.

You need to set up a set-driven key (SDK) to raise or lower the heel based on the value of the Heel attribute.

- 2. Select the left foot control oval and choose Animate → Set Driven Key → Set. Click Load Driver to load the control oval as the driver of this SDK.
- 3. Select lt_rev_heel and click Load Driven in the SDK window.
- 4. In the SDK window, select Heel as the driver attribute on the oval and the RotateZ attribute on the driven joint (Figure 9.90).
- 5. Make sure the Heel attribute is set to **0** and the Rotate Z for the lt_rev_ heel joint is also **0**. Click Key in the SDK window.
- 6. Set the Heel attribute to -5 and rotate the lt_rev_heel 45 degrees up in Z. Click Key in the SDK window. If you now select the foot control oval and change the Heel attribute to between -5 and 0, the heel will raise and lower. Make sure to set it back to **0** when you're done testing.
- 7. Select the lt_rev_ballFoot joint and click Load Driven in the SDK window. Keep If foot CNTRL's Heel as the driver. Select the RotateZ attribute for the driven joint (See Figure 9.91.)
- 8. Set Heel to **0** and rotate in Z for the rev_ballFoot to **0**. Click Key in the SDK window.
- 9. Set Heel to 5 and rotate in Z for the rev_ballFoot to 45 degrees. Click Key in the SDK window.
- 10. Set Heel to **10** and rotate in Z for the rev_ballFoot back to **0**. Click Key in the SDK window.
- 11. Test the Heel slider by sliding it to a value of 5, and you should see the back of the foot moving up, as shown in Figure 9.92. Set Heel to 10, and the heel comes back down.

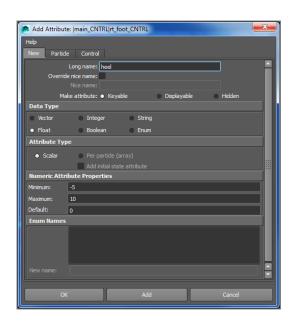


Figure 9.89

Create a new attribute for the foot control oval.

12. Finally, select lt_rev_toes and click Load Driven in the SDK window. Keep lf_foot_ CNTRL's Heel as the driver. Select Rotate Z as the driven channel for the lt_rev_toes joint in the SDK window. Figure 9.93 shows the SDK relationship.



Figure 9.90

The first SDK relationship for the reverse heel joint



Figure 9.91

The next SDK relationship for the reverse ballFoot joint

Figure 9.92

When the Heel attribute is at 5, the heel should raise up like shown.

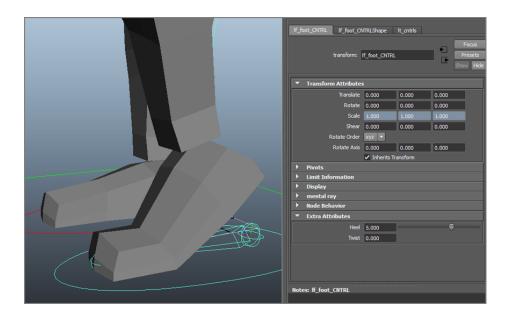




Figure 9.93
The final SDK relationship for the

reverse toe joint

- 13. Set Heel to **5** and make sure the lt_rev_toes joint's Rotate Z is **0**. Click Key in the SDK window.
- 14. Set Heel to **10** and rotate in Z the lt_rev_toes joint to **45** degrees. Click Key in the SDK window.
- 15. Test the Heel slider, and you'll see at a value of 10 that the foot is up and the toes are on the ground (see Figure 9.94).
- 16. Repeat steps 1 through 15 for the right foot. This time, you will use the rt_nodes in place of the lt_nodes you used earlier.

Make sure you're saving your progress as you go! When you're done, your rig should have some pretty handy controls. You will move the entire character using the main_CNTRL node. You can move the body itself while keeping sticky IK hands and feet by using the body_CNTRL node. And, of course, you control the arms/hands and legs/feet using the respective arm_CNTRL and foot_CNTRL nodes. All keyframes and animation should happen on these nodes only; you should not have to manipulate the joints or the IK handles directly at all. And that, in a somewhat confusing and long-winded nutshell, is the purpose of a character rig.

You can load the file character_rig_v01.ma from the Block_Man project to check your work or just to use as a rigged character for some animation fun!

For Further Study

Animation is as much a sport as it is an art, meaning practice makes you better. Use the rigged character from the end of this chapter to redo the walk animations from the beginning of the chapter. Also try different types and moods of walk. Can you animate a simple walk that looks happy and enthusiastic? Can you make that walk sad and lonely? Being able to convey emotion in your movement is key, and the more you try, the better you get.

Summary

In this chapter, you extended your experience with animation and learned about rigging techniques and automation. Starting with the simple Block Man, you learned how to set up a hierarchy for forward kinematics animation to create a walk cycle. Then, you used a skeleton to rig a hand for animation. Next, you learned how to bind the geometry of the hand to the skeleton using two different methods of smooth bind and also the interactive bind, as well as how to edit the binding. You also learned how to create an IK

system to drive the joints in the Block Man for an IK walk cycle animation. After that, you learned how constraints can be used in rigging and how to set up set-driven keys to create easy controls to animate the hand. Then, you put all these rigging tricks together to rig the wheels of the locomotive to automate the animation of that complex system with a single control based on the middle wheel. And finally, you tackled a pretty tough rigging assignment in rigging the Block Man even further to have some nice options for movement using controllers.

The true work in animation comes from recognizing what to do in the face of certain challenges and how to approach their solutions. Maya offers a large animation toolset, and the more familiar you become with the tools, the better you'll be able to judge which tools to use in your work. Don't stop with this chapter; experiment with the features not covered here to see what happens.

Animation is about observation and interpretation. The animator's duty is to understand how and why something moves and to translate that into their medium without losing the movement's fidelity, tenacity, or honesty.

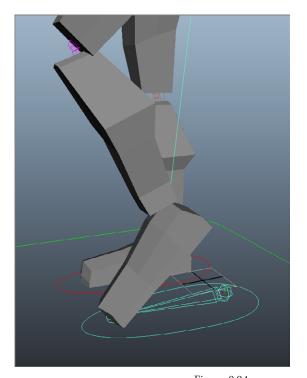


Figure 9.94
The foot position when Heel is set to 10



Autodesk® Maya® Lighting

Light shapes the world by showing us what we see. It creates a sense of depth, it initiates the perception of color, and it allows us to distinguish shape and form. For a scene to be successful in CG, these realities of light need to be reproduced as faithfully as possible. The trick is learning to see light and its astonishing effects on the world around us.

Learning Outcomes: In this chapter, you will be able to

- Understand basic concepts for setting up CG lighting
- Choose the appropriate Autodesk® Maya® light for a scene based on light attributes
- Control which lights illuminate certain objects through light linking
- Create mood and realism with raytraced shadows
- Illuminate and render a scene with mental ray® Physical Sun and Sky
- Produce special lighting effects with volumetric lighting, lens flare, and shader glow
- Practice setting up a basic lighting solution for the toy airplane, glass candle holder, and decorative box
- Use raytracing to cast shadows in your scene and use refractions to create a glass look
- Animate the attributes of a light and aim lights with the special manipulator

Basic Lighting Concepts

It's no surprise that lighting in Maya resembles actual direct-lighting techniques used in photography and filmmaking. Lights of various types are placed around a scene to illuminate the subjects as they would be for a still life or a portrait. Your scene and what's in it dictate, to some degree at least, which lights you put where.

Although it's easy to insert and configure lights, it's *how* you light that will make or break your scene. Knowing how to do that really comes only with a good deal of experience and experimentation, as well as a good eye and patience.

This chapter will familiarize you with the basic techniques of lighting a scene in Maya and start you on the road to finding out more.

What Your Scene Needs

Ideally, your scene needs areas of highlight and shadow. Overlighting a scene flattens everything and diminishes details. Figure 10.1 shows a still life with too many bright lights.

Similarly, underlighting your scene makes it muddy and lifeless, and it flattens the entire frame. Figure 10.2 shows the still life underlit. The bumps and curves of the mesh are hardly noticeable.

Like a good photographer, you want your image to have the full range of exposure. As in Figure 10.3, light and shadow complement each other and work to show the features of your surface.

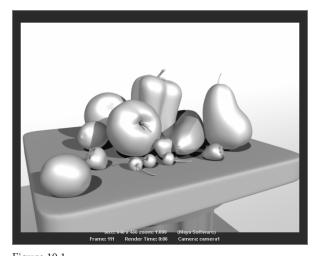


Figure 10.1 An overlit still life

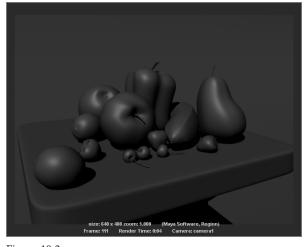


Figure 10.2 An underlit still life

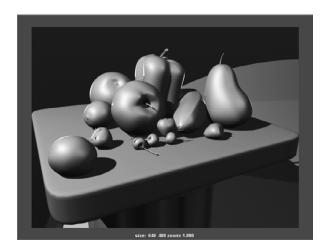


Figure 10.3 **Balanced lighting** creates a more interesting picture.

Three-Point Lighting

The traditional filmmaking and television approach to lighting is called *three-point lighting*. Three distinct roles are used to light the subject of a shot. More than one light can be used for each of the three roles, but the scene should seem to have only one primary (or key) light, a softer light to fill the scene, and a back light to pop the subject out from the background.

Three-point lighting ensures that the primary subject's features aren't just illuminated but featured with highlights and shadow. Using three directions and qualities of light creates the best level of depth. Figure 10.4 shows a schematic of a basic three-point setup.

Key Light

A *key light* is placed in front of the subject and off to the side of the camera to provide the principal light on the subject. Because it's usually off-center, the key light creates one side of brighter light, increasing the depth of the shot. This light also provides the primary shadows and gives the important sense of lighting direction in the shot.

Although it's possible for several lights to fulfill the role of key light in a scene—for example, three ceiling lights overhead—one light should dominate, creating a definitive direction. Figure 10.5 (left) shows the subject being lit by only a key light, although it's physically composed of two lights.

Figure 10.4 A three-point lighting schematic

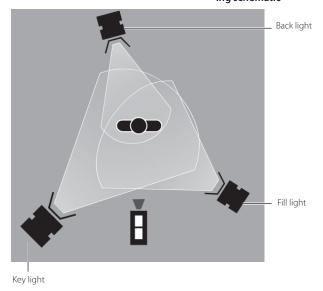
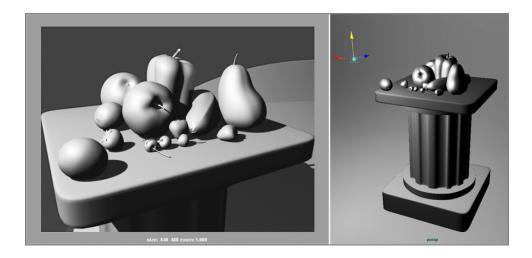


Figure 10.5 Key light only



The direction of the two lights remains the same, and one takes intensity precedence over the other and casts shadows. The effect creates a single key light, which produces a moody still life.

Try This

- 1. Open the scene file still_life_v01.mb from the Lighting project downloaded from the book's website.
- 2. In the cameral panel, press 7 for lighting mode. It should turn black; there are no lights.
- 3. Click Create → Lights → Point Light and, in the persp window, place it as your key light (Figure 10.5, right). Use the cameral panel to gauge how the lighting composition is working for optimum placement of the Point light.

Fill Light

A more diffused light than the key light, the *fill light* seems directionless and evenly spread across the subject's dark side. This fills the rest of the subject with light and decreases the dark area caused by the key light.

The fill light isn't usually meant to cast any shadows onto the subject or background itself and is actually used to help soften the shadows created by the key light. Figure 10.6 shows the still life with an added fill light. Notice how it softens the shadows and illuminates the dark areas the key light misses.

Typically, you place the fill light in front of the subject and aim it so that it comes from the opposite side of the key light to target the dark side of the subject. Even though the still life in Figure 10.6 is still a fairly moody composition, much more is visible than with only the key light in the previous figure.

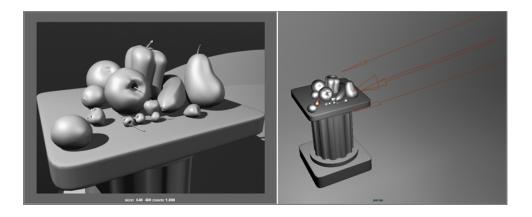


Figure 10.6 A fill light is now included.

Try This

- 1. In the existing scene that you started with a single Point light, choose Create → Lights → Directional Light. Where you place the light doesn't matter, but how you rotate it does.
- 2. Rotate the light so you get a lighting direction opposite to the direction of the key light from the Point light already in the scene.
- 3. With the Directional light selected, change its Intensity attribute in the Channel Box or the Attribute Editor from 1.0 to **0.5**. Use the cameral panel to see how the fill light is working.

Back Light

The back light, or rim light, is placed behind the subject to create a bit of a halo, which helps pop the subject out in the shot. Therefore, the subject has more presence against its background. Figure 10.7 shows how helpful a back light can be.

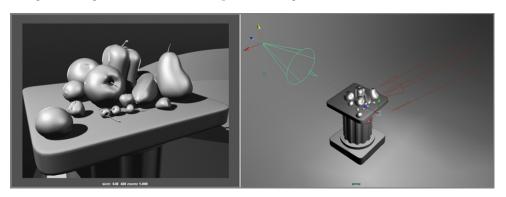


Figure 10.7 A back light makes the subject pop

right out.

The back light brings the fruit in this still life out from the background and adds some highlights to the edges, giving the composition more focus on the fruit.

Don't confuse the back light with the background light, which is used to light the environment itself.

Try This

- 1. In your current scene with the two lights, create a third light to be a Spot light.
- 2. You can use Move and Rotate to position the light to shine from behind the fruit, or you can use the special manipulator. To use that, press T with the Spot light selected. You will see two Move manipulators, one for the source of the light and one for the target.
- 3. Move the target to the front of the column stand, and move the target behind and slightly above the fruit (Figure 10.7, right). Use the cameral panel to see how the back light should be placed.

Using Three-Point Lighting

The three-point lighting system is used for the primary subject of the scene. Because it's based on position and angle of the subject to the camera, a new setup is needed when the camera is moved for a different shot in the same scene. Three-point lighting is, therefore, not scene specific but shot specific, as long as it does not break the overall continuity of the scene.

After the lighting is set up for the subject of a shot, the background must be lit. Use a directed primary light source that matches the direction of the key light for the main light, and use a softer fill light to illuminate the rest of the scene and soften the primary shadows.

Maya Lights

Six types of light are available in Maya: Ambient, Directional, Point, Spot, Area, and Volume. These lights are also used when rendering in mental ray. How you use each dictates whether they become key, fill, or rim lights. Each light can fill any of those roles, although some are better for certain jobs than others. The most commonly used light types for most scenes are Spot, Directional, and Ambient. All of these Maya lights render in Maya software as well as mental ray.

To create each light, choose Create → Lights and click the light type; that light will appear at the origin of your scene.

Common Light Attributes

Lights in Maya are treated like any other object node. They can be transformed, rotated, scaled, duplicated, deleted, and so forth, and they are visible as nodes in the Hypergraph and Outliner alongside other objects in the scene. Like any other node, lights have attributes that govern how they function. Figure 10.8 shows the Attribute Editor for a typical light.

When you select any light type and then open the Attribute Editor, you'll see the following attributes and options:

Type This drop-down menu sets the type of light. You can change from one light type to another (for instance, from Spot to Point) at any time; however, you may not keyframe changing a light's Type attribute.

Color This attribute controls the color cast by the light. The darker the color, the dimmer the light will be. You can use Color in conjunction with Intensity to govern brightness, although it's best to set the brightness of a light to Intensity only. You are able to keyframe Color easily by simply RMB+clicking the attribute name Color in the Attribute Editor and choosing Set Key. Alternately, you may RMB+click Color in the Channel Box and choose Key Selected from the context menu.

Intensity This attribute specifies how much light is cast. The higher the intensity, the brighter the illumination will be. You can also keyframe intensity by RMB+clicking the attribute just as with Color.

Illuminates By Default This check box deals with *light linking*, or the ability to illuminate specific objects with specific lights. Clearing this check box causes the light not to illuminate all objects by default, requiring you to link the light to objects you do want it to light. Keep this check box checked unless you're linking lights to specific objects. This chapter will briefly touch on light linking later.

Emit Diffuse and Emit Specular For all light types except the Ambient light type, these check boxes toggle on or off the ability to cast diffuse lighting or specular highlights on an object (see Figure 10.9). This is useful for creating specific lighting effects. For example, if lighting an object makes it too shiny, you can disable the specular emission from one or more of the lights on that object to reduce the glare.

Figure 10.8

A typical light's

Attribute Editor

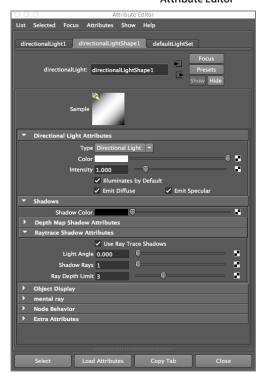
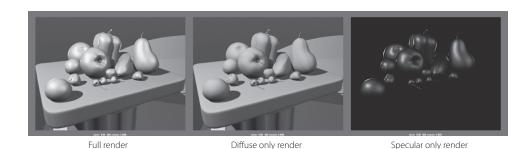


Figure 10.9 Lights can render diffuse or specular components if needed.



Light Types

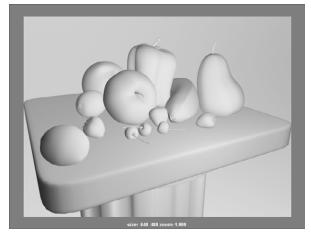
Beyond the common light attributes, each light type carries its own attributes that govern its particular settings. In the following section, open the scene file still_life_v01.mb from the Lighting project and create the light being described to see firsthand how that particular light affects the scene.

Ambient Lights

Ambient lights cast an even light across the entire scene. These lights are great for creating a quick, even illumination in a scene; but, as you can see in Figure 10.10, they run the risk of flattening the composition. They're perhaps best used sparingly and at low intensities as fill lights or background lights.

The Ambient Shade slider in the Attribute Editor governs how flat the lighting is. The lower the value, the flatter the lighting. Figure 10.11 shows the effect of two contrasting Ambient Shade settings.

Figure 10.10 **Ambient light**



Directional Lights

Directional lights cast a light in a general direction evenly across the scene (see Figure 10.12). These lights are perhaps second to Spot lights as the most commonly used light type. They're perfect for sunlight or general indoor lighting, for key lights, and for fill and back lights. They give an accurate sense of direction without having to emanate from a specific source.

Point Lights

A *Point light* casts light from a single specific point in space, similar to a bare light bulb. Its light is spread evenly from the emission point (see Figure 10.13).

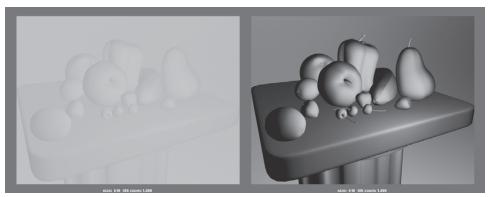


Figure 10.11

A low Ambient Shade setting flattens the image.

Ambient Shade = 0.1

Ambient Shade = 1.0

Using the Decay Rate drop-down menu in the Attribute Editor, you can set how a Point light's intensity diminishes over distance. With No Decay, the Point light illuminates an object far away as evenly as it does up close. This is the most common setting for most applications.

Setting Decay Rate to Linear, Quadratic, or Cubic requires you to increase the intensity level exponentially to compensate for the decay. You can use Decay Rate settings to illuminate nearby objects and to leave distant ones unaffected. In reality, lights have decay rates. But in CG, they don't really need to decay unless the falloff effect is needed, as shown in Figure 10.14. Clever lighting can easily avoid this cumbersome calculation.

Point lights are good for effects such as candlelight or setting a mood.

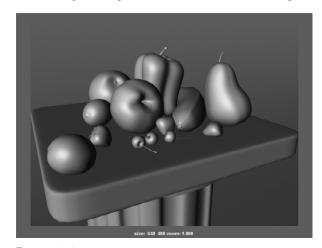


Figure 10.12

Directional light

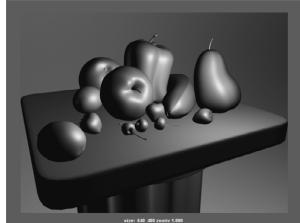


Figure 10.13

A Point light placed in the front right of frame

Spot Lights

Spot lights are arguably the most used lights in Maya because they can be used for keys, fills, or rims, and they're efficient.

Similar to Directional lights, Spot lights emphasize direction. But these lights emit from a specific point and radiate out in a cone shape, whereas Directional lights emit from an infinite source from a certain direction. As such, Spot lights can create a circular focus of light on the geometry much like a flashlight on a wall; Directional lights spread the light evenly. Figure 10.15 shows a Spot light on the still life.

The following attributes govern the behavior of Spot lights:

Decay Rate Specifies the rate at which the light's intensity falls off with distance, as with the Point light. Again, the intensity needs to increase exponentially to account for any decay.

Cone Angle Sets the width of the cone of light emitted by the Spot light. The wider a cone, the more calculation intensive it becomes.

Penumbra Angle Specifies how much the intensity at the edges of the cone and hence the circular focus dissipate. (See Figure 10.16.) A negative value softens the light into the width of the cone, decreasing the size of the focus; a positive value softens away from the cone.

Dropoff Specifies how much light is decayed along the distance of the cone. The higher the drop-off, the dimmer the light gets farther along the length of the cone. This effect is much better to use than a decay rate, and it gives similar results.

Most practical lights are created with Spot lights. For example, a desk lamp's light is best simulated with a Spot light. Spot lights are also the lights of choice to cast shadows. You'll find more on shadows later in the chapter. Remember, with a Spot light, you can press T for the special manipulator, allowing you to move the source and target of the Spot light easily to orient and place the light in your scene.



Figure 10.14 A Point light with a Decay Rate set

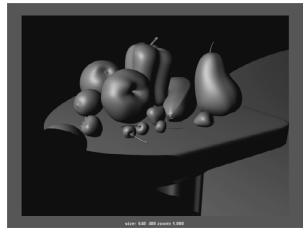


Figure 10.15 Using a Spot light

Penumbra = 0

Penumbra = -10

Penumbra = 10

Figure 10.16

The Penumbra Angle attribute controls the softness of the edge of a Spot light.

Area Lights

Area lights emit light from a flat rectangular shape only (see Figure 10.17). They behave similarly to Point lights, except they emit from an area and not from a single point. You can still set a decay rate, just as you can with Point lights. Area lights are the only lights whose scale affects their intensity. The larger an Area light, the brighter the light.

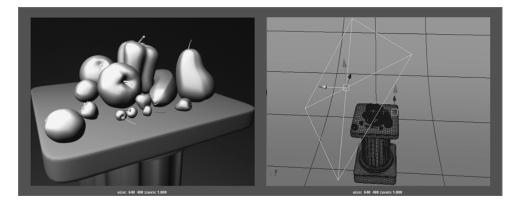


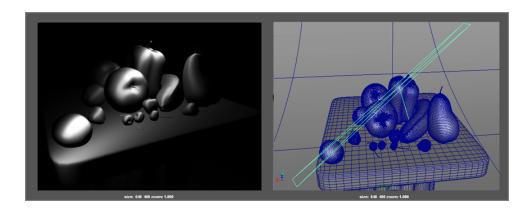
Figure 10.17

An Area light and its placement

Because you can control the size of the area of light being emitted, these lights are good for creating effects such as a sliver of light falling onto an object from a crack in a door (as in Figure 10.18), overhead skylights, or the simulation of large, diffused lighting fixtures such as overhead office lights. Use Area lights when you need to light a specific area of an object.

Additionally, Area lights give off a softer feel the larger they are. Often area lights give the best photographic light quality compared to the other light types, especially when used with mental ray rendering, and they create the most accurate-looking shadows, though at the expense of render times.

Figure 10.18 An Area light as a sliver and its placement



Volume Lights

Volume lights emit light from a specific 3D volumetric area as opposed to an Area light's flat rectangle (see Figure 10.19). Proximity is important for a Volume light, as is its scale.

A Volume light can have the following attributes:

Light Shape A Volume light can be in the shape of a sphere, a box, a cylinder, or a cone. You select a shape from the Light Shape drop-down menu.

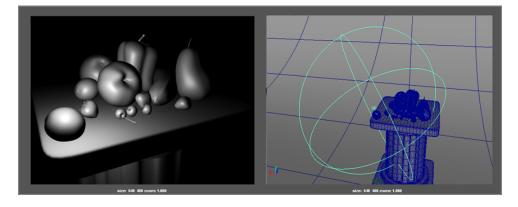
Color Range This section of attributes sets the color of the light using a built-in ramp. The ramp (from right to left) specifies the color from inside to outside. For instance, a whiteto-black ramp from right to left creates a white light at the center of the Volume light that grades down to black toward the outer edge.

Volume Light Dir This attribute sets the direction for the light's color range: Outward lights from inside out, Inward lights from the volume's edge into the center, and Down Axis lights as a gradient in an axis of the light.

Arc And Cone End Radius This attribute defines the shape for the volume.

Penumbra For cylinder and cone shapes, this attribute adjusts how much the light dims along the edge of its length.

Figure 10.19 A Volume light and placement



Use Volume lights when you need to control the specific area in which light is cast or when you need an object to move into and out of a particular area of light. Volume lights are not to be confused with *volumetric lighting*, such as the effect of a flashlight in smoke or headlights driving through fog. For volumetric effects, see the section "Volumetric Lighting" later in this chapter.

Lighting a Scene

Any of these aforementioned lights will work with Maya Software rendering and mental ray rendering. It's also best to start with just a couple types of light, such as Directional and Spot no matter which renderer you use, before turning to the more sophisticated types, such as Area and Volume.

Getting the essence of lighting is far more important in the beginning than understanding the nuances of all the attributes of a light. At first, limit yourself to Spots and Directional lights and try to avoid any Ambient light use.

Light Linking

You can control which lights illuminate which objects by using Maya *light linking*. However, by default, lights created in your scene illuminate all objects in the scene. The easiest way to create an exclusive lighting relationship is first to create a light and then to turn off the attribute Illuminates By Default in the light's Attribute Editor. This ensures that this light won't cast light on any object unless specifically made to do so through light linking.

- 1. Open the scene file still_life_linking_v01.mb from the Lighting project or use your own scene that already has at least three lights. Click the Render button () in the Status line to render a frame (Figure 10.20). There are no shadows; you'll deal with that in the next section.
- 2. Create a new Directional light and, in the Attribute Editor, turn off the check box Illuminates By Default (it's right below the Intensity slider). Set Intensity to 1 or more. You can see the placement of the light in Figure 10.21.
- 3. Render a frame, and you won't see any change. Adding a new light with Illuminates By Default disabled won't increase the light level in the scene.
- 4. To assign your new light to the objects you want to illuminate exclusively, choose Window → Relationship Editors → Light Linking → Light-Centric. This opens the Relationship Editor and sets it for light linking. Light-Centric means the lights are featured in the left side of the panel, as shown in Figure 10.21, and the objects in your scene that will be lit are on the right.

Figure 10.20
All of the scene's lights illuminate the

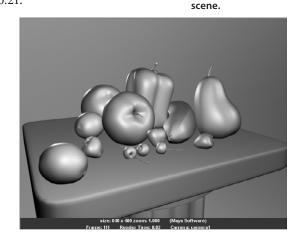
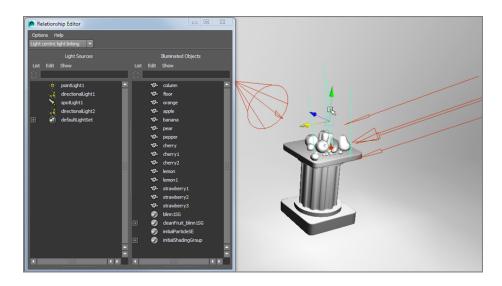


Figure 10.21

The Light Linking window and the newly added Directional light



- 5. Now, select the light you want to link (in this case, the directionalLight2 you just created) and the objects in the scene you'd like to link to (in this case, the apple and the pepper, as shown in Figure 10.22). Notice that no other objects on the right side of the Relationship Editor are selected; this means they will receive no illumination from this light source.
- 6. Render a frame, and you'll see that the objects you linked are lit by the new light. In this case, the apple and the pepper are brighter than the other fruit in the still life. (See Figure 10.23.)



Figure 10.22

Select the scene objects to link to the Directional light.

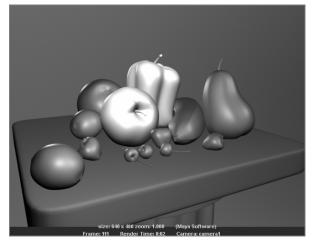


Figure 10.23

A linked light creates extra light for only the apple and the pepper behind it. The other objects aren't illuminated by that light.

When you're in Lighted mode (press 7 in the Shaded panel) and using the default Viewport 2.0 view panel display, linked lights aren't taken into account in that panel display. However, if you switch the view panel from Viewport 2.0 to High Quality Rendering (in the view panel's menu bar, choose Renderer → High Quality Rendering), you will be able to see light linking work in the view panel. Light linking works with both Maya Software rendering and mental ray rendering.

Adding Shadows

Don't be too quick to create an abundance of light in your scene to show off your models and textures. Shrouding objects in darkness and shadow is just as important as revealing them in light. A careful balance of light and dark is important for a composition. As Figure 10.24 shows, the realism of a scene is greatly increased with the simple addition of well-placed shadows. Don't be afraid of the dark. Use it liberally but in balance.

Creating Shadows in Maya

Lights can cast shadows in one of two ways, depending on how the scene is rendered: depth map shadows and raytraced shadows. Depth map shadows are faster; however, with computing being as fast as it is now, it's usually just best to use raytraced shadows because they are more accurate and usually look much nicer. We will use raytraced shadows throughout this book.

When you create a light in Maya, raytraced shadows are enabled by default for the light. However, since Maya Software rendering does not have raytracing enabled by default, your renders will not have any shadows at first. You will need to turn on raytracing or switch to mental ray rendering to see your raytraced shadows. You will explore raytracing and mental ray further in Chapter 11, "Autodesk" Maya® Rendering."

If you do not want to raytrace your shadows for whatever reason, you will need to enable shadow maps for your lights by clicking the Use Depth Map Shadows check box in the Shadows area of the light's Attribute Editor. Maya will then generate shadow maps

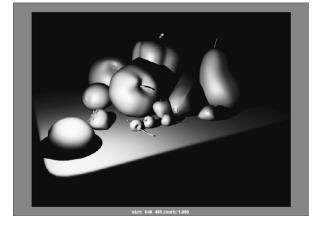
that locate where shadows fall by following the path of the light backward from the lighted object to the light itself. Shadow maps are generally fast because they do not need raytracing, but they are not very accurate.

On the other hand, *Raytracing* involves tracing a ray of light from every light source in all directions and tracing the reflection to the camera lens. Therefore, you can create more accurate shadows in your renders.

As you can see, depth map shadows are not as nice as raytraced shadows when you compare Figure 10.25 and Figure 10.26.

Figure 10.24

Darkness and shadow help add a sense of realism, depth, and mood to an otherwise simple still life.



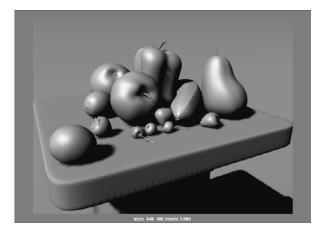


Figure 10.25 A light with depth map shadows renders faster, though not as detailed as raytracing.

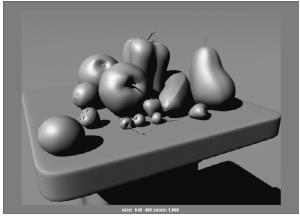


Figure 10.26 A light with raytraced shadows produces more detailed shadows.

Raytraced Shadows

When you create a light, the light's Use Ray Trace Shadows setting in the Attribute Editor (see Figure 10.27, toward the bottom of the Attribute Editor window) is turned on by default. If you are rendering with Maya Software, open the Render Settings window by choosing Window → Rendering Editors → Render Settings or by clicking the Render Settings icon (🕌) in the Status line; then enable the Raytracing check box under the Raytracing Quality heading. Let's add shadows to the scene using Maya Software rendering first.

- 1. In the file still_life_linking_v01.mb from the Lighting project, select the Point light. In the Attribute Editor, turn on Use Ray Trace Shadows under the Shadows → Raytrace Shadow Attributes heading (Figure 10.27, right). In this scene, raytrace shadows have been disabled on purpose. All new lights you create will automatically have raytraced shadows enabled.
- 2. Select the Directional light, and turn on Use Ray Trace Shadows.
- 3. Open the Render Settings window and enable Raytracing (Figure 10.27, left).
- 4. Render a frame, and you'll see the scene looks more natural now with shadows (Figure 10.28). Turning on raytracing has also enabled reflections, which you'll look at later this chapter. Notice how the brighter key light (Point light) casts darker shadows than the darker fill light (Directional). As a general rule, you don't want too many lights in the scene all casting shadows; that may confuse the composition.

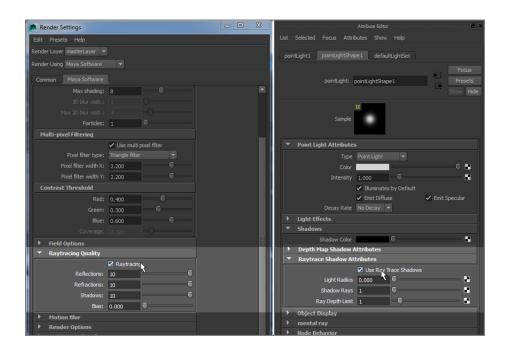


Figure 10.27

Make sure Use Ray Trace Shadows is checked on for the light (right) and enable Raytracing for the Maya Software renderer in the Render Settings window (left).

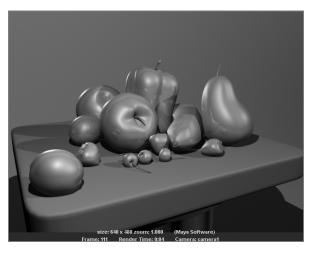


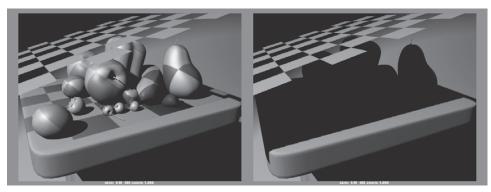
Figure 10.28

Raytraced shadows add a lot to the scene.

For an object that has a transparency map applied to its shader, however, only ray-traced shadows can cast proper shadows. On the left of Figure 10.29 is a plane with a mapped checkerboard transparency casting a raytraced shadow over the still life. On the right is the same light using shadow maps instead of raytraced shadows.

Figure 10.29

Only raytraced shadows work with transparencies.



Raytraced shadow

Shadow map shadow

Controlling Shadows per Object

To better control your lighting, you can specify whether an object can cast and receive shadows in Maya. For example, if you have geometry casting light in front of a shadow but you don't want it to cast a shadow, you can manually turn off that feature for that object only.

To turn off shadow casting for an object, follow these steps:

- 1. Select the foreground lemon in the still_life_linking_v01.mb scene and open the Attribute Editor.
- 2. In the Render Stats section is a group of check boxes that control the render properties of the object, as shown in Figure 10.30 (left). Clear the Casts Shadows check box. If you don't want the object to receive shadows, clear the Receive Shadows check box.
- 3. Render a frame, and you'll see the lemon has no shadows on the column's tabletop; compare the render in Figure 10.30 (right) with Figure 10.28.

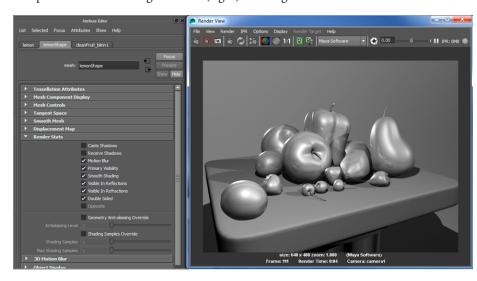


Figure 10.30

You easily can set whether an object casts or receives shadows in the **Attribute Editor** (left). The lemon does not cast a shadow anymore (compare with Figure 10.28).

Figure 10.31

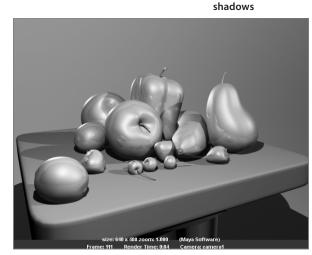
Regular raytraced

Raytracing Soft Shadows

One interesting feature of shadows is that they can diffuse or soften as the shadow falls farther away from the object casting the shadow. For example, hold a pen on its end on your desk and notice how the pen's shadow gets softer or fuzzier the farther it is from where the pen meets the table. This small detail can greatly enhance the reality of any render. To use soft shadows, follow these steps:

1. Open the still_life_shadows_v01.mb scene from the Lighting project. Check out the render settings. Raytracing is already enabled for the key and fill lights as well as for the renderer.

- 2. Render a frame and notice that the shadows are sharp. Save that image into the render buffer by clicking the Keep Image () icon in the Render View settings. In Figure 10.31, you can see the still life rendered with sharp, raytraced shadows.
- 3. Select the Point light and open the Attribute Editor. Under the Shadows → Raytrace Shadow Attributes heading, set Light Radius to 0.2, as shown in Figure 10.32.

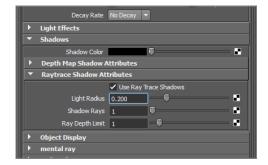


4. Select the Directional light and set its Light Angle option to 4.0. A Directional light's attribute is called Light Angle instead of Light Radius, but it does the same thing. Now render a frame and compare it to the previous frame you rendered. The shadows are soft but very noisy, especially on the background (Figure 10.33). Save your image in the render buffer.

The Directional light had a much higher light radius than the Point light because its shadows react differently. The Point light's distance from the subject plays a big part in how the light affects the shadows from the subject, so it is more sensitive to Light Radius values than is the non-position-dependent Directional light.

- 5. Now you'll mitigate the noise. Select the Point light and in, the Shadows → Raytrace Shadow Attributes heading, set Shadow Rays to 16. Select the Directional light and set its Shadow Rays value to 12.
- 6. Render the frame, and your shadows should resemble the ones in Figure 10.34. The higher your Light Radius value, the higher you need to set your Shadow Rays setting to compensate for noise.

Figure 10.32 Setting the Light Radius value





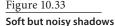




Figure 10.34 A higher Shadow Rays setting makes for cleaner soft shadows.

Notice how the shadows soften more toward the edge of the shadow; the shadow is still sharp at the point of contact. This gives a much nicer feeling of depth to the scene. There is an increase in render time—it's important to evaluate how much softening you need so you don't overdo the look or increase the render time too much.

Using soft shadows was easy! Make sure you enable raytracing in Render Settings, of course. These soft raytraced shadows work both in Maya Software rendering as well as in mental ray rendering.

mental ray Lighting

mental ray lighting and rendering opens up a large range of possibilities within Maya. As with all rendering, lighting plays a primary role. I'll cover mental ray rendering more in the next chapter; however, because rendering and lighting go hand in hand, it's tough to ignore it in this chapter. This section is a primer on mental ray light functionality.

Open the Render Settings window by choosing Window → Rendering Editors → Render Settings. If you don't see the mental ray option in the Render Using drop-down menu, you need to load the plug-in. Choose Window → Settings/Preferences → Plug-in Manager to open the Plug-in Manager. Make sure Mayatomr.mll (or Mayatomr.bundle on a Mac) is checked for Loaded as well as for Auto Load to ensure that it loads by default.

Two additional functions that mental ray brings to the Maya table are caustics and global illumination (GI). Caustics is the scattering of light reflections off and through semitransparent objects, such as the light that shines on the ceiling above an indoor pool or the sunshine at the bottom of an outdoor pool. Global illumination is the effect of light reflected from one object to another. For example, if you place colored spheres inside a gray box and shine a light into the box, the walls and floor of that box pick up the color

of the spheres. The light from the spheres reflects onto the walls and tints them with the spheres' color. Furthermore, the light from the floor of the box bounces and helps illuminate the undersides of the balls.

For example, Figure 10.35 shows a scene file that has a dozen or so glass spheres inside an enclosed box. The box has four holes in the top, and two Spot lights with shadows turned on are positioned outside the box, shining in through the holes. Figure 10.35 shows a typical Maya Software render. The spheres are visible and reflect the environment, but overall they are dark, with the rest of the box in shadow and no bounced light.

However, when rendering through mental ray for Maya (see Figure 10.36), the light that enters the box bounces around the scene and illuminates the other spheres. The color of the spheres also colors the area immediately around them because of GI. Additionally, the light shines through the semitransparent spheres and casts caustic highlights on the floor. If you are reading a digital version of this book, you can see the color bouncing around in Figure 10.36.

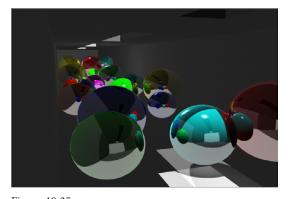


Figure 10.35 The Maya Software render of the box of spheres scene



Figure 10.36 The mental ray for Maya render of the scene

Image-Based Lighting

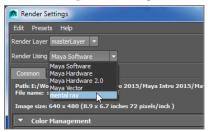
mental ray also gives you image-based lighting (IBL) in Maya. This method of lighting uses an image, typically a high dynamic range image (HDRI), to illuminate the scene using Final Gather or GI. Final Gather is a form of global illumination that relies on direct as well as indirect illumination. Direct illumination calculates the amount of light coming directly from lights in the scene and renders the result. However, it misses an important aspect of real-life lighting: diffuse reflections of light. Indirect illumination happens when light bounces off objects in a scene in order to reach and therefore light the rest of the scene—that is, diffuse reflections. Final Gather is typically a faster way than GI to get indirect illumination in a scene.

I'll briefly touch on Final Gather here as we explore Physical Sun and Sky lighting in the next section. However, I'll cover both IBL and Final Gather in depth in Chapter 11, when you light the toy plane and decorative box model (with displacement maps for box details as well).

mental ray Physical Sun and Sky

In Physical Sun and Sky lighting, mental ray for Maya creates nodes in your scene to simulate an open-air sunlight effect for your scene lighting. It's a quick way to create a

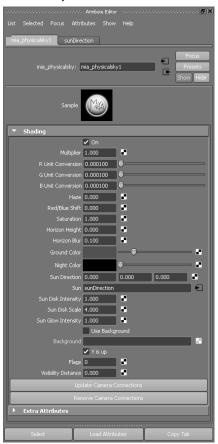
Figure 10.37 Enable mental ray rendering in Render Settings.



nice-looking render. You'll place the textured red wagon into an open scene and apply a Physical Sun and Sky (PSAS) in the following exercise.

Make sure your project is set to the Lighting project from the web page and use the scene file WagonSunlight_v01.ma from the scenes folder to follow along with these steps:

- Figure 10.38
- The mia_physicalsky1 node



- 1. The camera for this scene is already set up in the persp panel. Open the Render Settings window and switch to mental ray rendering in the Render Using pull-down menu at the top of the window, as shown in Figure 10.37.
- 2. In the Render Settings window, click the Indirect Lighting tab. Under the Environment heading, click the Create button next to Physical Sun And Sky. The
 - Attribute Editor opens for the mia_physicalsky1 node you just created, as shown in Figure 10.38.
 - 3. Render a frame in the Perspective window and compare it to Figure 10.39. It's a bit bright, but there's something very nice about this render. Notice that the metal handlebar takes on a good look right off the bat.

The render is too bright, and the wood slats for the railings shouldn't be reflective. Your render should also show that the fireengine red of the wagon is a bit washed out.

- 4. Let's first address the wood railings. Open the Hypershade, and double-click the first Wood shader. Change Reflectivity to 0.1 from 0.5. Repeat for the second Wood shader.
- 5. In the Render View window, click and drag a box region around the wood railings, as shown in Figure 10.40. Click the Render Region button () to render that part of the frame only.
- 6. The wood looks much better now, so let's move on to the brightness of the scene. Open the Render Settings window, and go to the Indirect Lighting tab. Click the Input arrow next to the Physical Sun And Sky attribute's button (Figure 10.41). Doing so opens the Attribute Editor for the Physical Sun and Sky.
- 7. Change Multiplier to **0.5** from 1.0, and re-render the frame. The brightness comes down nicely, and the wagon is less blown out than in your previous renders. See Figure 10.42.



Figure 10.39 The first PSAS render doesn't look too shabby.

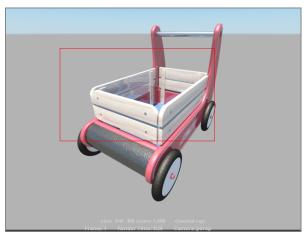


Figure 10.40 Render this region to check the reflection in the wood railings.

8. Let's play with the direction of the sun. Look in your persp panel, and you see a Directional light sitting smack in the middle of the scene. Maya uses this light to set the direction of the sunlight, and it doesn't contribute to the lighting of the scene in any other way. Only its rotation is important to PSAS. Its intensity, color, and other attributes are irrelevant. In this scene, the Directional light

is called sunDirection and is in the wagon behind the third texture placement node, as you can see in Figure 10.43. The light is currently pointing sunlight almost straight down, as if the sun is high in the sky. It seems as if it's about noon. RotateX for the sunDirection light is -75.



Figure 10.41 Click to open the attributes for the daylight system.



Figure 10.42 Bringing down the brightness of the sun



Figure 10.43

The sunDirection light adjusts only the direction of the sunlight in the scene.

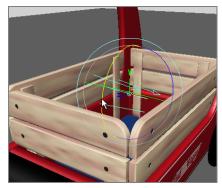


Figure 10.44 Angle the sun farther away from the camera.

- 9. That third texture placement node is getting in the way. To turn off its display easily, in the persp view panel, click Show → Texture Placements to uncheck it. The green box disappears in this view until you turn it back on through the Show menu. Now that you can see the Directional light better, you see that the sun is pointed slightly toward the back of the wagon. Rotate the light so that it's angled away from the camera even more, as shown in Figure 10.44. The sunDirection's RotateX should be about -25.
- 10. Render the frame, and you'll see quite a difference in the scene. Maya automatically adjusts PSAS settings to make the render appear as if the sun is about to set. This is because of the new angle. The scene (shown in Figure 10.45) not only shows a new lighting direction (the shadows fall toward the back and are longer) but also is a darker and warmer light, as if it were
- 11. Angle the sunDirection light even more (so RotateX is at about -2.5), almost parallel with the ground plane, and render a frame. Now the sun has all but set, and it's dark dusk, just before night falls on the scene. See Figure 10.46.
- 12. Rotate the sunDirection even more so that it's at a RotateX of about 15. It's now dark, and you can barely make out the wagon. (See Figure 10.47.) It's time to go home before your mom gets mad. Remember, you can adjust the overall brightness of the scene by adjusting the mia_physicalskyl node attributes, as you did in steps 6 and 7.



mid-to-late afternoon.

Figure 10.45 Now it's late afternoon.



Figure 10.46 It's getting late; you should get home before nightfall.





Figure 10.47 Ooh, it's dark-Mom is gonna be mad.

Figure 10.48 The sun also rises!

You can add lights to the scene as well; you aren't limited to the system's results. For example, Figure 10.48 shows a render of the wagon with the sun beginning to rise behind it, making the foreground a bit dark. Figure 10.49 shows the same render but with an added Directional light (with an Intensity of just 0.25) pointing toward the front of the wagon. This helps define the front of the wagon, hinting that there is another light source, perhaps a porch light behind the camera.

The PSAS system can get you some fairly nice results quickly. Keep experimenting with different sunDirection angles as well as the attributes for the system to see what results you can get for your scene. Add lights to create areas of detail in your model.

Unbeknownst to you, when you invoked the PSAS system, Maya turned on Final Gather in the mental ray settings, as well as adding a few nodes to the scene including a sky environment that surrounds your scene. This enabled indirect lighting to work in the scene. To get rid of the PSAS, you must go to the Render Settings window, and in the Indirect Lighting tab where you first turned on PSAS, you must click the Delete bar.

You'll explore Final Gather in the next chapter. At the end of that chapter, though, I'll introduce HDRI and image-based lighting for a fairly photo-real rendering of the toy plane and decorative box.

Figure 10.49 Adding a light to illuminate the front of the wagon in this sunrise scene



This is the perfect time for a break, so save your work (as if I have to tell you that at this point!), go grab some iced tea, and rest your eyes for a bit. In the next section, I'll go over various special lighting effects before returning to the wagon.

Lighting Effects

In CG, you must fake certain traits of light in the real world. Using certain methods, you can create smoky light beams, glowing lights, and lens flares. Although some of these effects fall under the domain of rendering and shader tricks, they're best explored in the context of lighting because they're created by light in the real world.

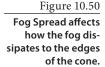
Volumetric Lighting

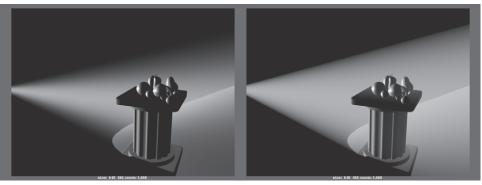
How do you create an effect such as a flashlight beam shining through fog? This lighting effect is called volumetric lighting, and you can use it to create some stunning results that can sometimes be time-consuming to render.

You can't apply volumetric effects to Ambient and Directional light types. To add a volumetric effect to any of the other types of lights, select the light and, in the Attribute Editor under the Light Effects section, click the checkered Map button to the right of the Light Fog attribute. This creates a new render node that appears in the Hypershade window. After you click the Map button, the Attribute Editor takes you to the lightFog node.

Maya handles volumetric lights by attaching a lightFog node to the light. The Color and Density attributes under this node control the brightness, thickness, and color of the fog attached to that light. Furthermore, in the light's Attribute Editor, you can control the fog with the Fog Spread and Fog Intensity settings. Fog Intensity increases the brightness of the fog, and Fog Spread controls how well the fog is defined within its confines. For example, a Spot light with fog shows the fog in its cone. Figure 10.50 shows how Fog Spread affects the conical fog shape.

To remove a fog effect, RMB+click the Light Fog label in the light's Attribute Editor and choose Break Connection from the context menu.





Fog Spread = 0.5

Fog Spread = 2.0

If you want the rays of light within the fog to cast shadows, check Use Depth Map Shadows for the light. You'll have to increase the depth map resolution for a higher-quality image.

Lens Flare

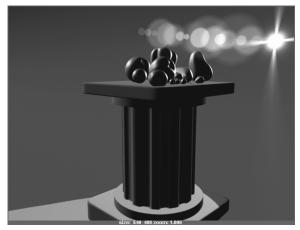
Lens flare and light glow, as illustrated in Figure 10.51, mimic the real-world effect created when light strikes a lens or when the light source is visible in the frame. The flare is created when the light hits the lens at a particular angle and causes a reflection of itself in the optics of the lens.

Figure 10.51

Light glow and lens flare turned on for the back light

To enable a light glow, under the Light Effects section in the light's Attribute Editor, click the checkered Map button next to the Light Glow attribute to create an OpticalFX node that appears in the Hypershade. The Attribute Editor shifts focus to that new node, which controls the behavior of the light glow and lens flare. The OpticalFX node contains the following attributes and settings:

Glow Type Setting this attribute specifies the kind of glow: Linear, Exponential, Ball, Lens Flare, and Rim Halo. These define the size and shape of the glow from the light.



Halo Type Specifying a halo creates a foggy halo around the light in addition to the glow. You can find controls for the halo in the Halo section in the Attribute Editor.

Star Points Setting this attribute specifies the number of star points the glow generates.

Rotation Setting this attribute rotates the orientation of the star points.

Radial Frequency Used in conjunction with the Glow Radial Noise attribute (see the next item) in the Glow Attributes section, this attribute defines the smoothness of any added glow noise.

Glow Radial Noise Setting this attribute adds noise to the glow effect, creating light and dark patches within the glow for a more random look, as shown in Figure 10.52.

Glow Color Setting this attribute specifies the color of the glow.

Glow Intensity and Spread Setting these attributes specifies the brightness and thickness of the glow and how well it fades away.

To turn on a lens flare along with the light glow, click the Lens Flare check box at the upper right in the Attribute Editor for OpticalFX. The attributes in the Lens Flare section control the look of the flare.

Glow Radial Noise attribute

Figure 10.52



Glow Radial Noise = 0

Glow Radial Noise = 0.5

Light glows and flares can be highly effective in scenes, adding credibility to the lighting, but they're often misused or, worse, overused in CG. Used sparingly and with subtlety, lens flares can go a long way toward adding a nice touch to your scene.

Shader Glow Effects

To create a glowing effect, it's sometimes better to place a glow on a geometry's shader instead of the light. Because a light must be seen in the shot and pointed at the camera to see any light glow and flare, a shader glow is sometimes more desirable. This process will composite a glow on the object that is assigned the Glow shader to simulate a volumetric light, such as a street lamp on a foggy night. Shader glows have far less render cost than true volumetric lights.

Try This To light a still-life scene, follow these steps:

- 1. Open the still_life_v03.mb file in the Lighting project on the web page. Create a Spot light, place it over the still life, and aim it directly down onto the fruit, as shown in Figure 10.53. Turn on Use Depth Map Shadows for the light and set Resolution to 1024. Set Penumbra Angle to 10 and Intensity to 1.5. Press 7 for Lighted mode in the Cameral view panel to see how the light is being cast.
- 2. The Spot light provides the practical light in the scene. You'll place a bare bulb on a wire directly above the fruit. Create a NURBS sphere, and position it right over the fruit but in the frame for cameral to see. In the Render Stats section of the Attribute Editor for the sphere, turn off Casts Shadows.
- 3. Create a long, thin cylinder for the light bulb's wire and position it as if the bulb were hanging from it, as shown in Figure 10.54. Turn off Casts Shadows for the cylinder as well.

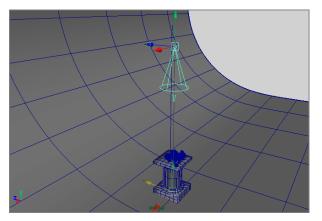




Figure 10.53

Aim a Spot light down toward the fruit.

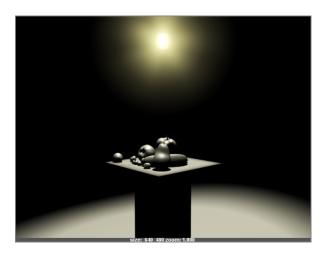
Figure 10.54

Place a bulb on a wire over the fruit pedestal.

- 4. Create a black Phong E shader to assign to the cylinder.
- 5. Create a Phong shader for the bulb and assign it to the NURBS sphere. Set its Color to a pale, light yellow, and make it about 50 percent transparent.
- 6. Select the Spot light and set its Color to the same yellow. You can do this easily through the Color Chooser. With the shader for the bulb selected, open the Color Chooser by clicking the pale yellow color you just made. Click the right arrow to place the yellow color in the swatches to the right of the main color swatch, or RMB+click any of the swatches.
- 7. Pick the Spot light and click its Color attribute to set the Color Chooser to that color. Click the yellow swatch you created to get the same color on the light. For detail's sake, make the light's color less saturated. Click Accept to close the Color Chooser window.
- 8. To make the shader glow for the bulb, open the Hypershade and select the bulb's Phong material. In the Attribute Editor's Special Effects section, drag the Glow Intensity slider from 0 to 1.0. If you render the frame, you'll see that the glow isn't quite enough to make a convincing light bulb. In the Hypershade, select the shader-Glowl node; this node controls all the glows in the scene.
- 9. Set Quality to **0.1**. In the Glow Attributes section, set Glow Intensity to **6.0**, set Glow Spread to **0.5**, and set Glow Radial Noise to **0.2**.

The scene file still_life_v04.mb from the web page contains the full scene for your reference. See Figure 10.55 for the final result.

Figure 10.55 The bare light bulb over the still life is created with a shader glow.



Assembling and Lighting a Scene

In this section, you'll set up and light a simple scene with the toy plane and the decorative box so that you can render them together. Furthermore, in the next chapter, you'll start using displacement maps for detail on the box, and you'll use HDRI, mental ray, and Final Gather to get the most from lighting and rendering. The lighting you'll start with will be basic for now. In the following exercise, you'll create a basic lighting setup for the scene and get a direct lighting solution first. In the next chapter, you'll expand on this lighting using mental ray and HDRI.

Set your current project to the Plane project, which you should have already downloaded from the book's web page. This is important to make sure Maya finds all the project files it needs, so it bears repeating: Set your project to Plane! Did you set your project yet?

How about now?

Assembling the Scene

To begin lighting, open the planeLightingScene_v01.mb scene file from the Scenes folder of the project. This is a simple scene with a camera in place and a corner made up of a redwood tabletop and two white walls:

- 1. Import the assets you need, starting with the toy plane. Select File → Import and, in the Options section on the right of the Import dialog window, turn off Use Namespaces, set Resolve to Clashing Nodes, and leave With set to the filename. This makes sure that Maya doesn't append any additional names to original objects being imported from this file. Next choose planeShading_v01.ma from the Scenes folder.
- 2. Open the Outliner and select the top node of the toy plane (planeGroup). Go into the four-view layout so you can see all four view panels. Select one of the view

panels and select Panels → Perspective → renderCam. This way you can work in the persp view panel and see how it affects the camera view from which you'll be rendering.

Next you will bring in the decorative box from the previous chapters. This time, instead of importing, you will use referencing. This allows you to import a file into your current scene. However, with a reference, any time you adjust the original file, it will automatically update that model in the scene it has been referenced in. You'll see this in action in the next chapter.

- 3. Choose File → Create Reference. On the right side of the Reference dialog window, check the Use Namespaces box to turn it on. This will place the filename you are referencing in front of the object names in the Outliner.
- 4. Navigate to your Plane project. In the Scenes folder, select boxDetail01.mb to bring in the latest decorative box scene from Chapter 7, "Autodesk" Maya Shading and Texturing," as a reference. This is actually the same scene file as boxTextures03.mb in the Decorative_Box project that you completed in Chapter 7, but cleaned up a bit and placed in the Plane project folders for simplicity's sake. The texture files you need are also copied into the Plane project's Sourceimages folder.
- 5. Rotate the box to 200 degrees in the Y-axis and position the box next to the toy plane to match Figure 10.56.

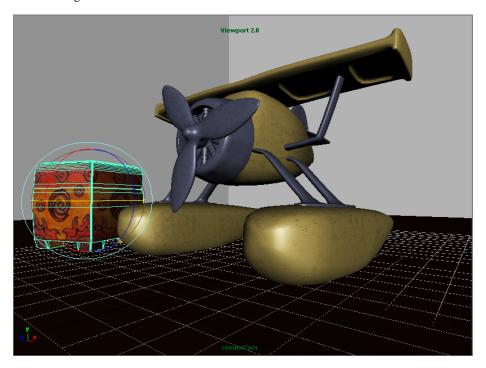
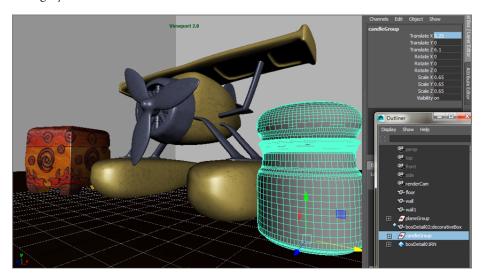


Figure 10.56 Orient and place the decorative box next to the plane to match this position.

- 6. You'll now import the glass candle holder you created in Chapter 5 using NURBS techniques. There's no need to use reference like you did with the decorative box, so select File → Import and turn off Use Namespaces, set Resolve to Clashing Nodes, and leave With set to the filename, just like in step 1.
- 7. Navigate to the candleholder project and, in the Scenes folder, select and import the candleModel_v03.mb file.
- 8. In the Outliner, select the candle's top node (candleGroup) and move it to the front side of the toy plane, near to the camera. You will also have to scale the candleGroup to **0.65** to match the position and size in Figure 10.57. Don't worry, the candle holder is all gray for now. Save!

Figure 10.57 Import and place the candle to match this layout.



Creating the Lights

Save your file and compare it to planeLightingScene_v02.ma or skip to the following section. Here you will lay out lighting using the three-point lighting system discussed earlier.

- 1. In the renderCam view panel, press 6 for Texture mode and then 7 for lighted display mode. Using the persp view panel, create a Spot light; in the Attribute Editor, set Cone Angle to **70.0** and Penumbra Angle to **10.0**. Place and orient the Spot light to be the key light, matching the position and orientation shown in Figure 10.58.
- 2. Create a Directional light and place that as shown in Figure 10.59. Set Intensity to **0.6**.
- 3. Create a rim light with a Spot light with an Intensity value set to **0.7**; place it as shown in Figure 10.60. Set Penumbra Angle to 10.0. Try using the special manipulator for the Spot light by pressing T. This way you can place the source and the target of the light to more easily position it.

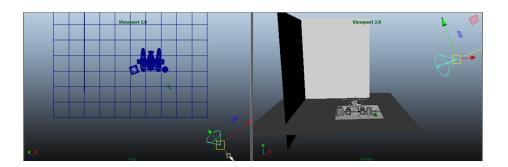


Figure 10.58

Create a Spot light and place it as shown.

Figure 10.59

Create and place a Directional light as shown here.

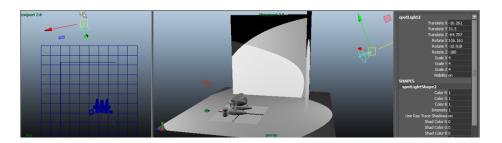


Figure 10.60

Create a Spot light for a rim light and place it as shown here.

4. In the renderCam view panel, render a frame (by clicking the sicon in the Status line); you should see something like what is shown in Figure 10.61. You need to create a simple glass shader for the candle, and there are no shadows or reflections yet, but it's a start. You'll enable raytracing in the next section to make a glass shader and render shadows.

Reflection and Shadows

Save your file and compare it to planeLightingScene_v03.ma or skip to the following section. Here you will make rendering reflections and shadows possible.

1. Open the Render Settings window () and click the Maya Software tab. In the Raytracing Quality section, click to enable Raytracing. If you render your scene now, you'll notice shadows in the image; however, the rim light has disappeared. That is because it is behind the back wall and is being blocked. You need to turn off shadows for that rim light in the next step.

A simple render of the plane



- 2. Select the Spot light that is your rim light (behind the back wall). In the Attribute Editor, under Shadows → Raytrace Shadow Attributes, uncheck the Use Ray Trace Shadows box to turn off those shadows.
- 3. If you render the frame now, it should resemble the image in Figure 10.62. It's a basic look, but it's getting nicer now that you see reflections in the objects (except the candle holder) and shadows.
- 4. Now let's soften the shadows. Select the first Spot light you created (the key light) and, in the Shadows → Raytrace Shadow Attributes section, set Light Radius to 9.0 and Shadow Rays to 12.





- 5. Render a frame, and the shadows look nicer and more real. You can experiment with the light's Light Radius and Shadow Rays values to dial in the shadows to your liking. The sharper the shadow, the smaller and the closer the source of the light will appear.
- 6. However, the reflections in the toy plane make the plastic look too shiny. Select the body of the plane in the persp view panel and open the Hypershade window. In the Hypershade, click the Graph Materials On Selected Objects icon () to bring up the plane's yellow Blinn shader in the Work Area.
- 7. Click the yellow shader and, in the Attribute Editor, under the Specular Shading section, set Eccentricity to **0.425**, Specular Roll Off to **0.525**, and Reflectivity to **0.25**.
- 8. Select the gray Blinn shader and repeat the same attribute values as step 7.
- 9. In the Hypershade, click the phongFloor shader and set its Cosine Power to 30 and Reflectivity to **0.15**. Render, and your frame should look more like Figure 10.63.

You can see that simple additions of shadow and reflections helped this render look nicer.



Figure 10.63 Reducing the glossy

look of the plastic toy plane and the wooden tabletop

Making Glass

Now turn your attention to the glass candle holder and the candle inside. You can continue with your own scene or load the scene file planeLightingScene_v04.ma to skip to this point. You will create a glass shader and a simple shader for the candle in the following steps:

1. In the Hypershade, create a new Phong shader and name it candleGlass.

- 2. Open the Outliner. MMB+click the plus sign to the left of the candleGroup node to expand all the child nodes beneath it. Select the glassJar and glassLid objects; in the Hypershade, RMB+click the new candleGlass Phong shader and choose Assign Material To Selection.
- 3. In the candleGlass shader, move the Transparency slider so that the black color swatch turns almost pure white. This will make the shader almost entirely transparent.
- 4. Set Cosine Power to 60 to make the specular highlights tighter and more glossy looking. Render a frame, and you'll notice the jar is now clear glass with a gray candle inside. You may also notice some strange vertical bars between the glass and the gray candle (Figure 10.64).

This phenomenon occurs when two surfaces penetrate each other. Since Maya is interpreting the geometry of the glass and the candle inside slightly differently during render time, this penetration may occur. You can simply reduce the size of the candle inside ever so slightly. However, you will address this issue when you render this scene with mental ray in the next chapter, so you can leave it be for now.

- 5. There is a gray ring inside the lid of the jar that should be a white plastic. In the Hypershade, create a new Blinn and give it a white color. Set Reflectivity to 0.1. Call this shader whitePlastic.
- 6. In the Outliner, select the plasticSealer object under the jarLid group and assign it to the whitePlastic shader.
- 7. Make another Blinn shader and give it a dark red color. Set Eccentricity to **0.125**, Specular Roll Off to **0.85**, and Reflectivity to **0.1**. Name it candleWax.



Figure 10.64

Rendering a glass look for the candle jar shows a little bit of penetration with the gray candle inside. That's okay for now.

- 8. In the Outliner, select the candle object grouped under the jarCandle group and assign it to the candleWax shader.
- 9. Finally, create a dark gray Lambert called blackWick and assign it to the wick object in the Outliner.
- 10. Render your scene and have a look. The candle is looking nicer, but the glass is missing something still. You need to create refractions in the glass of the candle holder.
- 11. Select the candleGlass Shader and, in the Attribute Editor, scroll down to and open the Raytrace Options section. Check the Refractions box to turn on refractions and set Refractive Index to 1.1. For refractions to show up, you will need to have a Refraction Index value other than 1. The lower or the higher than 1.0 that you set this value, the more distorted the refraction will be. Small deviations from 1.0 give you the best results; going too high or too low may look super weird.
- 12. Before you render, go to the Render View window first (if you closed Render View after your last render, choose Window → Rendering Editors → Render View). Click and drag a box around the candle in the image of your last render and then click the Render Region icon () to just render that part of the frame to save time. Your glass render should look much more like glass now (Figure 10.65).

The render is not looking too shabby, but when you tackle this scene again with mental ray in the next chapter, you will see how powerful mental ray rendering can be. In Chapter 11, you'll add more texture maps to add carving detail to the decorative box, create mental ray specific shaders for the objects in the scene, and enable Final Gather to use an HDR image to light the scene to take it to a new level. Woo!

You can check your work against the scene file planeLightingScene_v05.ma in the Scenes folder of the Plane project.



Figure 10.65

The glass looks much nicer with refractions enabled.

Further Lighting Practice

Lighting professionals in the CG field are called on to find the most efficient way to light a scene and bring it to the peak of its beauty. Again, this comes only from experience. The best way to become a crackerjack lighting artist is to spend years honing your eye and practicing the latest procedures, such as HDR lighting.

The file still_life_v01.mb in the Lighting project on the web page contains the scene of the still life with no lights, so you can play with lighting and shadow methods as well as light linking to create some extra focus on some parts of the frame. The file still_life_ v02.mb contains the same scene but with three-point lighting already set up.

Notice in the still_life_v02.mb file that two lights make up the key light (spotLight1 and spotLight2). One light makes up the fill light (directionalLight1), and two lights (spotLight3 and spotLight4) make up the back light.

For practice, download some models from the Internet and arrange them into your own still-life scenes to gain more lighting experience. Set up scenes, time the rendering process, and try to achieve the same lighting look using faster lighting setups that may not be as taxing on the renderer. Also, try taking pictures of situations and trying to match the lighting in the photo.

Try setting up simple scenes. Start with an indoor location that is lit by a single light bulb. Then, try the same scene in the following locations to expand your lighting repertoire:

- A photography studio
- Outside in the morning on a bright summer day
- Outside at dusk in the fall
- · Outside at night under a street lamp
- Inside on a window ledge
- At the bottom of a closet lit by a nearby hallway light

Tips for Using and Animating Lights

When you're lighting a scene, invoking a lighting mode in your perspective or camera view panel will give you great feedback regarding the relative brightness and direction of your lights. Most consumer graphics cards can handle a maximum of eight lights in Lighted mode; some professional cards can handle more.

As you've seen, you invoke Lighted mode by pressing 7 on your keyboard (not through the number pad on the side). You must first be in Shaded mode (press 5) or Texture mode (press 6) to be able to press 7 for Lighted mode. Remember that Lighted mode displays linked lights as if they're lighting the entire scene. This can cause some confusion, so it's wise to take notes on any light linking in your scene.

The Maya IPR renderer is also useful when lighting a scene. This almost-real-time updating renderer will give you a high-quality render of your scene as you adjust your lights. Chapter 11 will explore the IPR renderer.

Animating a Light

Any attribute of a light can be animated in the same way that you animate any other object attribute. You can't, however, animate a light's type. To edit a light's animation, you need only select the light and open the Graph Editor to access its keyframes. You can set keyframes on Intensity, Penumbra Angle, Color, and so on, within the Channel Box or the Attribute Editor. RMB+click the name of the attribute and choose Key Selected from the context menu.

By animating a light's intensity, you can simulate the real-world appearance of a light turning on or off. To turn on a light, create a quickly increasing curve so that its brightness arcs up slowly at first before climbing to full brightness. This animation mimics the way real lights turn on and off better than simply enabling or disabling them in your scene.

When you animate the color of a light or a shader, you set keyframes for the color's RGB values as three separate keyframes. The Graph Editor shows a separate curve for the red, green, and blue channels of color when you animate a light's color. You can set all three keys at once by RMB+clicking the Color attribute in the Attribute Editor and choosing Set Key from the context menu, as shown in Figure 10.66.

In addition, lights can be animated to be moved, scaled, and rotated like any other object. For further study, try animating the lighting for the simple scene(s) you set up to practice lighting from the previous section. Try creating animated lights to simulate a candle illuminating your scene, a campfire, or the flashing emergency lights you would find in your average space-station airlock.

Figure 10.66 Set a key for the light.



Using the Show Manipulator Tool for Lights

As you saw earlier in the chapter, an easy way to manipulate lights is to use their special manipulator. For example, pressing T or choosing Modify → Transformation Tools → Show Manipulator tool with a Spot light selected gives you two Translate manipulators in the view panel, as shown in Figure 10.67, and displays the icon () below the Tool Box.

This allows you to move the source or target of the light to aim it better. By clicking the cyan circle that appears below the source's Translate manipulator, you can toggle through a number of manipulators to adjust the Spot light's settings, such as cone angle (two clicks clockwise) and penumbra angle (three clicks clockwise). Figure 10.68 shows the manipulator for cone angle.

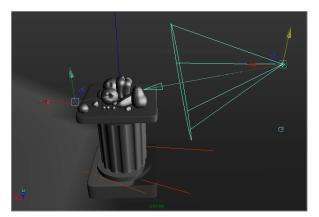


Figure 10.67

Using special manipulators to place and orient the Spot light

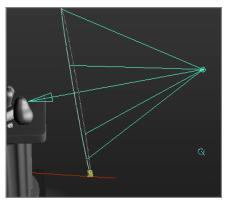


Figure 10.68

Adjusting the cone angle interactively with the special manipulator

Source and Target Translate manipulators are available for all light types through the Show Manipulator tool as well.

Summary

This chapter explored lighting in Maya, beginning with basic concepts that included the three-point lighting technique. You then learned about the different lights in Maya, how they work, and how you can use light linking to control your scene better. Shadows are an important part of lighting and were covered in this chapter, followed by a quick exploration of the Physical Sun and Sky system with mental ray and then lighting effects such as lens flare and light glows. You then created simple lighting for the toy plane, candle holder, and decorative box for a still rendering. You used raytracing to enable reflections and shadows as well as to create a glass look with refractions. Finally, you learned how to begin animating lights for your scenes.

Lighting is truly the linchpin of CG; it can make or break a scene. As you'll see in the next chapter, lighting goes hand in hand with rendering and shading, and the more you understand about all three functions, the better your scenes will look. Don't be afraid to experiment with lighting and shading schemes on all your projects.



Autodesk® Maya® Rendering

Rendering is the last step in creating your CG work. It's the process by which the computer calculates the surface properties, lighting, shadows, movement, and shape of objects, and it saves a sequence of images. Although the computer does all the thinking at this point, you still need to set up your cameras and the render to get exactly what you want.

This chapter will show you how to render your scene using the Autodesk® Maya® Software renderer and mental ray® for Maya, as well as how to create reflections and refractions. In this chapter, you'll use a wine bottle and a still life from previous chapters, and you'll animate a camera to render a sequence.

Learning Outcomes: In this chapter, you will be able to

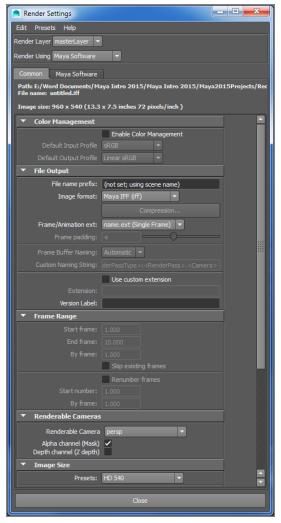
- Set up your scenes for output through rendering
- Choose resolutions, image formats, and other settings for rendering
- Create and edit cameras
- Render with motion blur and differentiate between the types of motion blur
- Set up and use mental ray
- Render in layers and compose them back together
- Set render layer overrides settings for specific needs
- Use Final Gathering in mental ray renders for indirect lighting workflows
- Use Ambient Occlusion to create detail in crevices and for contact shadows
- Apply an HDRI map as a lighting source
- Apply a displacement map to create finer detail in a model

Rendering Setup

When your lighting scene from the previous chapter is complete, you've had a celebration smoothie for your hard work, and you're ready to start a render, you'll need to set up how you want it rendered. Although this is the last part of the CG process, from now on you should be thinking about rendering throughout your production. When you create models and textures with the final image in mind and you gear the lighting toward showing off the scene elegantly, the final touches are relatively easy to set up.

Decide which of the render engines included with Maya you'll use: Maya Software, Maya Hardware (or Hardware 2.0), mental ray for Maya (the most popular of the renderers standard with Maya 2015), or Maya Vector. Each engine has its own particular workflow and can yield entirely different results, although mental ray and Maya Software

Figure 11.1 The Render Settings window



are close in appearance if you don't use the special features of mental ray. The choice of a rendering method depends on the final look you want and sometimes on the number of machines and licenses with which you can render. Maya Software rendering comes with an unlimited number of licenses, which means you can render on any machine you have (with Maya installed), although you can work with the Maya application only on as many machines for which you have licenses. There are also third-party developers in the CG field that have created other render engines that plug right into Maya, such as V-Ray® for Maya, Maxwell®, and Pixar's RenderMan® for Maya.

No matter which renderer you use, the lighting and general setup are fairly common across the board. Even then, it's a good idea to choose your render engine as you begin creating your scene. It's best to begin with basic mental ray to pick up the fundamentals of lighting, texturing, and rendering before you venture into mental ray special features or try other renderers.

Regardless of the type of render, you need to specify a set of common attributes in the Render Settings window. Choose Window - Rendering Editors - Render Settings to open the Render Settings window. Figure 11.1 shows the Render Settings window for the Maya Software renderer and its tabs: the Common tab and the Maya Software tab. You use the options in this window's Common tab to set up all your rendering preferences, including the resolution, file type, frame range, and so forth.

The Common tab contains the settings common to all the rendering methods, such as image size. The Maya Software tab gives you access to render-specific attributes, such as quality settings, raytracing settings, motion blur, and so on.

If you switch the Render Using pull-down menu from the default Maya Software setting to mental ray, you'll notice several tabs: Passes, Features, Quality, Indirect Lighting, and Options. These tabs give you access to all the settings for the powerful mental ray renderer.

You may notice that some of the render engines like mental ray may not show up in the Render Settings window right away. In this case, the renderers' respective plug-ins must be loaded first. I'll discuss this further in the section "mental ray for Maya" later in this chapter.

Choosing a Filename

Rendered images are identified by a filename, a frame number, and an extension, in the form <code>name.#.ext</code>. This is an example: stillife.0234.tif.

In the File Name Prefix text box, enter the image sequence name. If you don't enter anything in this text box, Maya automatically names your rendered images after your scene file (stillife in the example). This is the preferred naming convention; using it, you can immediately identify the scene file from which a particular image file was rendered.

In the Frame/Animation Ext drop-down list box, select <code>name.#.ext</code> to render out a sequence of files. If you leave this setting at the default of <code>name.ext</code>, only a single frame will render, no matter what the animation range is in the Time slider.

Name.#.ext is perhaps the most commonly used convention, as opposed to name.ext.# or name.#, because it allows you to identify the file type easily in Windows. Although Mac OS X isn't as picky about the order of the number and extension, most Mac compositing software applications (such as After Effects and Shake) want filenames that end in the three-letter extension. Therefore, it's best for both Mac and Windows users to employ the name.#.ext format.

The extension portion of the image name is a three-letter abbreviation for the type of file you're writing to disk to ensure that you can identify the file type.

Image Format

In the Image Format drop-down list box, select the type of image file you want to render. Maya will add the appropriate extension to the filename.

The image format you choose depends on your own preference and your output needs. For example, JPEG files may be great for the small file sizes preferred on the Internet, but their color compression and lack of alpha channel (a feature discussed later in this chapter) make them undesirable for most professional CG work beyond test renders.

It's best to render a sequence of images rather than a movie file for two reasons. First, you want your renders to be their best quality with little or no image compression.

Second, if a render fails during a movie render, you must re-render the entire sequence. With an image sequence, however, you can pick up where the last frame left off.

The best file type format to render to is TIFF. This format enjoys universal support, has little to no compression, and supports an alpha channel. Almost all image-editing and compositing packages can read Targa- and TIFF-formatted files, so either is a safe choice most of the time. For more on image formats, see Chapter 1, "Introduction to Computer Graphics and 3D."

Frame Range

Maya defaults to the frame range 1–10, which you may likely need to change to render your entire sequence. You must choose a naming convention other than Name (Single Frame) to access the frame range. Enter the Start Frame and End Frame attributes for part of the sequence or the entire sequence.

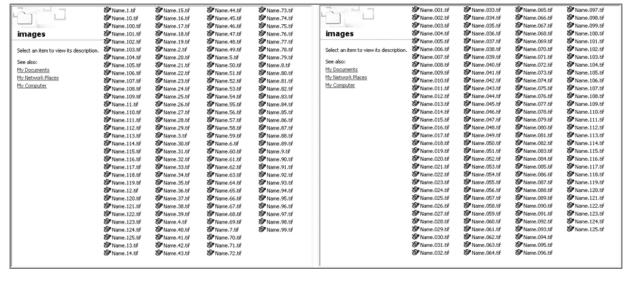
The By Frame attribute specifies the intervals at which the sequence will render. For example, if you want to render only the odd-numbered frames, set the Start Frame attribute to 1, and set the By Frame attribute to 2. If you want to render only even-numbered frames, set Start Frame to 2, set By Frame to 2, and so on. Typically, you leave By Frame set to 1 so that Maya renders each frame.

The Frame Padding attribute and slider have to do with the way an operating system, such as Windows or Mac OS X, orders its files by inserting leading zeros in the frame number. For example, if Frame Padding is set to 4, the filename contains three leading zeros; therefore, frame 8 is name.0008.tif as opposed to name.8.tif (which is set to a padding of 1).

Large sequences of files are easier to organize if they all have a frame padding of at least 3. Figure 11.2 shows an image sequence without padding and with padding. The files without padding aren't shown in numeric order.

Figure 11.2

Images rendered without frame padding (left). Frame padding makes file sequences easier to organize (right).



Camera and Channels

Under the Renderable Cameras heading, you choose the camera to render and enable the Alpha and Depth channels.

Image files are composed of red, green, and blue channels. Each channel specifies the amount of the primary additive color (red, green, or blue, respectively) in the image. (See Chapter 1 for more on how computers define color.) Some file formats save a fourth channel, called the *alpha channel*. This channel defines the image's transparency level. Just as the red channel defines how much red is in an area of the image, the alpha channel defines how transparent the image is when layered or composited on another image. If the alpha channel is black, the image is perfectly see-through. If the alpha channel is white, the image is solid and opaque. The alpha channel is also known as the *matte*. An object with some transparency like tinted glass will render with a gray alpha channel, as shown in Figure 11.3.

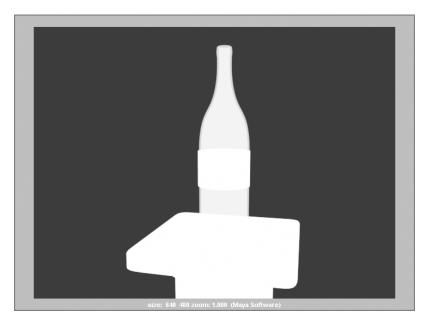


Figure 11.3

This wine bottle's transparency renders with a gray alpha channel.

The alpha channel can be displayed in the Render View window. As discussed later in this chapter, your test renders also display in this window.

To view an image's alpha channel in the Render View window, click the Display Alpha Channel icon (). To reset the view to RGB (full-color view), click the Display RGB Channels icon ().



Figure 11.4 Output an alpha channel.

Most renders have the alpha channels selected in the Render Settings window, so leave the Alpha Channel (Mask) check box selected at all times, as shown in Figure 11.4. Note, however, that JPEG, Graphics Interchange Format (GIF), and Windows bitmap files don't support alpha channels, regardless of whether the Alpha Channel (Mask) check box is selected.

Setting Resolution

The Width and Height attributes set the pixel size of the image to be rendered, a.k.a. the image resolution. In the Image Size section of the Render Settings window, you can select a resolution from the Presets drop-down list. The commonly used resolution for professional broadcast is 1920×1080 High Definition, which appears as HD 1080 in the Presets list. To composite Maya CG into a home-shot digital video (DV) movie, you use the standard DV resolution of 720×480 to render your scene, but you must enter that resolution manually in the Width and Height fields. (For more on resolutions, see Chapter 1.)

The Device Aspect Ratio and Pixel Aspect Ratio attributes adjust the width of the image to accommodate certain professional output needs; you do not need to adjust them here.

Make sure your Pixel Aspect Ratio attribute is set to 1 before you render, especially HD resolutions, unless you need to render CCIR 601/Quantel NTSC or DV for television needs; otherwise, your image may look squeezed or widened compared to any live-action footage you use to composite.

The higher your resolution, the longer the scene will take to render. With large frame sequences, it's advisable to render tests at half the resolution of the final output or less to save time. In addition to turning down the resolution for a test, you can use a lowerquality render.

Selecting a Render Engine

Maya allows you to select a render engine in the Render Settings window. Although mental ray for Maya is most commonly used, the other rendering methods give you flexibility in choosing a final look for your project.

Maya Software

Maya Software, the default software rendering method, can capture just about everything you want in your scene, from reflections to motion blur and transparencies. You can use the software rendering method in a couple of ways.

USING RAYTRACING

Raytracing, a topic introduced in Chapter 10, "Autodesk" Maya" Lighting," is used to incorporate two optical effects into a rendering that the default software rendering method can't handle. *Raytracing* traces rays of light from each light source to every object in the shot and then traces the light's reflection from the object to the camera's lens. This allows true *reflections* and *refractions* to appear in the render as well as highly defined shadows (for more on shadows, see Chapter 10).

True Reflections True reflections occur when every object in the scene is viewed in a reflective surface, as a reflection, of course. You can also have objects with reflections explicitly turned off through the Render Stats section in the Attribute Editor in case you don't want a particular reflection, which is common. Although it's possible to simulate reflections in Maya Software using *reflection maps*, true reflections can be generated only through raytracing. (In Chapter 7, "Autodesk" Maya" Shading and Texturing," the TableLamp project shows how to apply reflection maps.)

Refractions *Refractions* occur when light bends as it passes through one medium into another medium of different density. For example, a pencil in a glass of water appears to be broken. Refraction can also be turned on or off explicitly through Render Stats for objects that you don't want to see refracting through another clear object.

You saw in the previous chapter that raytracing is also a vital component of mental ray for Maya as well as the Maya Software renderer.

As soon as raytracing is enabled, any reflective surface receives a true reflection of the objects and environment in the scene. Even objects with reflection maps reflect other objects in addition to their reflection maps. For more on reflection maps, see the section "Reflections and Refractions" later in this chapter.

RENDER QUALITY

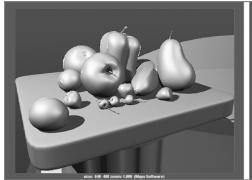
With software rendering, the render quality depends most noticeably on *anti-aliasing*. Anti-aliasing is the effect produced when pixels appear to blur together to soften a jagged edge on an angled line. Increasing the anti-aliasing level of a render produces an image that has smoother angles and curves. The Render Settings window contains presets that specify this level and a few others to set the quality of your render. Follow these steps:

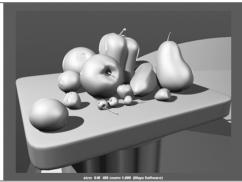
- 1. In the Render Settings window, make sure Maya Software is selected in the Render Using drop-down list and click the Maya Software tab.
- 2. In the Anti-aliasing Quality section, select either Preview Quality or Production Quality from the Quality preset drop-down list.

Figure 11.5 shows the fruit still life from Chapter 10 rendered with the Preview Quality preset and the same image with the Production Quality preset. Of course, the higher the quality, the longer the render will take.

Figure 11.5

With Preview Quality (left), the edges of the fruit are jagged. With Production Quality (right), the jaggedness is gone.





Maya Hardware and Hardware 2.0

The hardware rendering method uses your graphic card's processor to render the scene. Hardware renders are similar to what you may see when you play a 3D video game.

This method results in faster render times, but it lacks some of the features and quality you get from a software render. In Figure 11.6, the first image shows a wine bottle as rendered through hardware. The render time is blazingly fast, but the quality suffers. The second image shows the software render of the same frame. Hardware rendering becomes a good way to test-render a scene, although only certain video cards fully support Maya Hardware rendering.

Figure 11.6

The hardwarerendered wine bottle (left) lacks some subtleties. The software-rendered wine bottle shows better specularity and surface detail (right).





To use the Maya Hardware renderer, in the Render Settings window, make sure Maya Hardware is selected in the Render Using drop-down list. To specify hardware quality, select a level from the Number Of Samples drop-down list on the Maya Hardware tab.

mental ray for Maya

mental ray for Maya has become a standard for rendering through Maya, supplanting the Maya Software renderer because of its stability and quality results. The mental ray for

Maya rendering method can also let you emulate the behavior of light even more realistically than the other rendering methods, as you saw in the previous chapter's Physical Sun and Sky exercise. Based on raytracing, mental ray takes the concept further by adding photon maps to the light traces. That is, it projects photon particles from lights and records their behavior and trajectory. The end result allows the phenomena of light *caustics* and *bounce*, also known as *global illumination*.

The mental ray for Maya renderer can be an advanced and intricate rendering language with shaders and procedures all its own. This chapter briefly covers one of the popular mental ray methods called Final Gather using HDR image based lighting. To use mental ray, you still need to be experienced with the basics of lighting and rendering. At its base, mental ray will give you results similar to those of Maya Software when all of the bells and whistles of mental ray are not enabled. So it's often best to just use mental ray throughout.

Maya Vector

Vector rendering lets you render your objects with an illustrated or cartoon look. You can render "ink" outlines of your characters to composite over flat-color passes. Figure 11.7 shows the fruit still life rendered with Maya Vector. This rendering method is different from the Maya toon-shading feature, which is briefly covered in Chapter 12, "Autodesk" Maya* Dynamics and Effects."

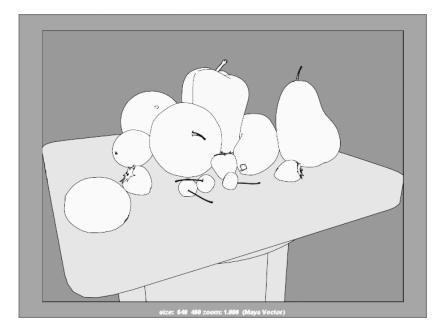
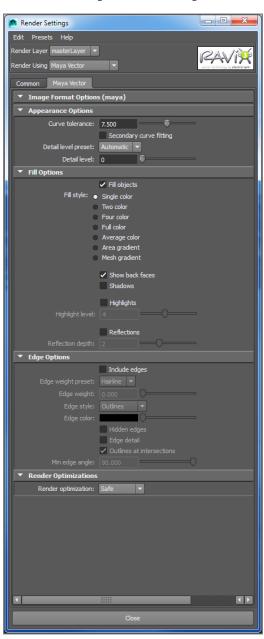


Figure 11.7
The fruit still life as a vector render

Maya Vector can output animated files in Adobe Flash format for direct use in web pages and animations, as well as Adobe Illustrator files and the usual list of image formats. To specify the attribute settings for Maya Vector, you use the Maya Vector tab in the Render Settings window (see Figure 11.8).

Figure 11.8 The Maya Vector rendering settings



In the Fill Options section, click the Fill Objects check box, and select the number of colors for each object to set the look of the render. If you want the renderer to include an outline of the edges of your geometry, in the Edge Options section, click the Include Edges check box and set the line weights.

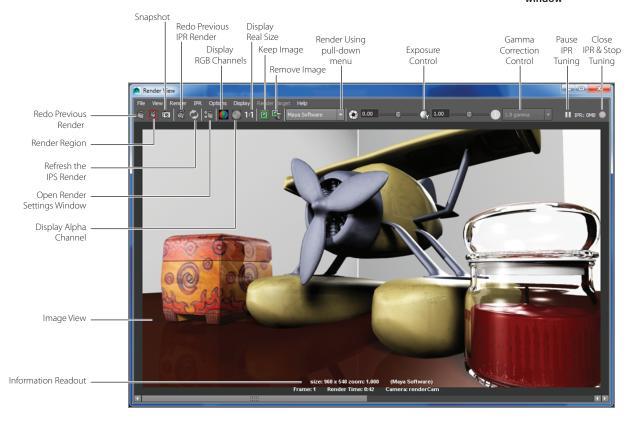
Previewing Your Render: The Render View Window

The Render View window automatically opens when you test-render a frame, as you've already seen in your work through this book. To open it manually, choose Window → Rendering Editors → Render View. Your current scene renders in the Render View window. Figure 11.9 shows the names of the icons in this window, and below we explore what the more important ones do.

Redo Previous Render Renders the last-rendered view panel.

Render Region Renders only the selected portion of an image. To select a portion of an image, click within the image in the Render View window and drag a red box around a region.

Figure 11.9 The Render View window



Open Render Settings Window Opens the Render Settings window (also known as the Render Globals window in previous versions of Maya).

Display RGB Channels Displays the full color of the image.

Display Alpha Channel Displays only the alpha matte of the render as a black-and-white image.

Display Real Size Resets the image size to 100 percent to make sure the image displays properly. When the Render View window is resized or when you select a new render resolution, the image renders to fit the window, and the image is resized if needed. If your render looks blocky, make sure the Render View window is displaying at real size before adjusting the options in the Render Settings window.

Render Using Pull Down Menu Lets you select the rendering method. This is the same as selecting it in the Render Settings window.

Exposure and Gamma Correction Controls These sliders allow you to adjust the brightness of the rendered image for viewing purposes. The first slider controls the overall brightness of the image while the second slider (Gamma Control) effectively runs the relative brightness of the lights and darks in the image, which in essence controls the contrast.

Information Readout At the bottom of the Render View window is a readout of information about the frame rendered. This information tells you the resolution, renderer used, frame number, render time, and camera used to render. This readout is a huge help in comparing different render settings and different frames as you progress in your work, especially when you keep images in the buffer (as explained later).

Saving/Loading an Image

Although you typically use the Render View window to test a scene, you can also use it to save single frames by choosing File – Save Image to save in any image formats supported in Maya. Likewise, choose File – Open Image to display any previously rendered image file in this window. If your task in Maya is to create a single frame, this is the best way to render and save it.

Keep/Remove Image

The Render View window is a prime place to see adjustments to various parts of your scene. You can store images in its buffer by clicking the Keep Image icon. When you do, a scroll bar appears at the bottom of the window, and you can scroll through any saved images. This is handy for making a change, rendering it, and scrolling back and forth between the old saved image and the new render to make sure the change is to

your liking. You can store a number of images in the buffer. For a faster way to preview changes, use IPR rendering, as discussed next.

IPR Rendering

As you saw in Chapter 7, a fast way to preview changes to your scene is to use Maya Interactive Photorealistic Rendering (IPR). After you IPR-render a view panel, specify the region you want to tune by dragging a box around that area of the image in the Render View window. Maya updates that region every time you make a shader or lighting change to the scene. Figure 11.10 shows the decorative box as an IPR mental ray render as the color and specular levels are being fine-tuned against a wireframe snapshot of the model.

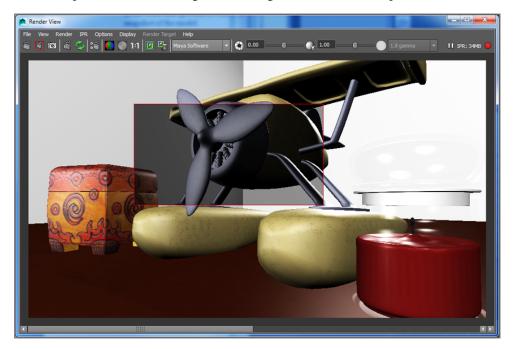


Figure 11.10

IPR rendering lets you fine-tune your textures and lighting with near-realtime feedback.

IPR is perfect for finding just the right lighting and specular levels. It will, however, register raytracing elements such as refractions and true reflections only if the scene is set to render through mental ray for Maya. Overall, IPR quality is fairly close to that of a full render while still allowing you to watch your tuning in near real time.

Setting the Default Renderer

By default, whenever you launch Maya, it sets Maya Software as the renderer of choice. To change this to mental ray, for example, in the main menu bar select Window - Settings/ Preferences - Preferences. In the Preferences window, select Rendering in the list on the

left side, and set the Preferred renderer drop-down menu to mental ray and click Save. From hereon, new scenes will default to using mental ray.

Reflections and Refractions

As you saw in Chapter 7, creating a "faked" reflection with a map for an object is pretty simple. To generate true reflections without the use of maps, however, you'll need to manually enable raytracing in Maya Software or use mental ray for Maya, which has raytracing enabled by default. With raytracing, Maya or mental ray reflects any objects in the scene that fall in the proper line of sight. All you need to do is use shaders that have reflection attributes such as Phong or Blinn. Note that Lambert shader does not reflect.

Figure 11.11 **Enabling raytracing** in Maya Software rendering

Raytraced Reflections

Rendering reflections using mental ray is simple. Just assign shaders to your objects that

Field Options Bias: 0.000

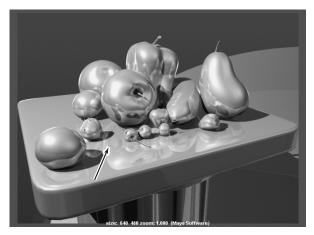
have a Reflection attrbiute such as Phong or Blinn. If you prefer not to use mental ray however, you must enable raytraced reflections with Maya Software rendering. First, make sure to use a material that has a Reflections attribute (for example, Phong or Blinn). Then, open the Render Settings window and choose Maya Software as your renderer. On the Maya Software tab, click the Raytracing check box in the Raytracing Quality section. (See Figure 11.11.)

The sliders control the quality of the render by specifying how many times to reflect or refract for any given object. Setting Reflections to 2, for example, enables an object's reflection in a second object to appear as part of its reflection in a third object.

Figure 11.12 Reflections set to 1 (left): Reflections set to 2 (right)

The first image in Figure 11.12 shows the still life reflecting onto the surface of its table. In this case, Reflections is set to 1. If you increase Reflections to 2, however, you'll see the reflections of the pieces of fruit in each other also reflecting in the surface of the table.





Notice the difference in the reflections of the fruit in the table between the two renders. Raytraced reflections can consume valuable render resources and time, so it's a good idea to make your scene efficient.

You can control the number of reflections on a per-object basis as opposed to setting limits on the entire scene through the Render Settings window. To access a shader's reflection limits, select the shader in the Hypershade and open the Attribute Editor. In the Raytrace Options section, drag the Reflection Limit slider to set the maximum number of reflections for that shader. The lower value (either this value or the Reflections value in the Render Settings window) dictates how many reflections are rendered for every object attached to that shader. The default shader reflection limit is 1, so make sure you change the Reflections value as well as each shader's value if you want more than one level of reflection.

Furthermore, you may not want some objects to cast reflections in a scene with raytraced reflections. To specify that an object doesn't cast reflections, select the object in a Maya panel and open the Attribute Editor. In the Render Stats section, clear the Visible In Reflections check box.

Rendering Refractions

Refractions are also a raytraced-only ability. Refractions require that an object be semitransparent so that you can see through it to the object (or objects) behind it being refracted. To control refractions, use the shader.

To enable refractions in either Maya Software or mental ray rendering, select the appropriate shader in the Hypershade and open the Attribute Editor. In the Raytrace Options section, click the Refractions check box. Now you need to set a refractive index for the shader and a refraction limit, similar to the reflection limit.

The refractive index must be greater or less than 1 to cause a refraction. Typically, a number within 0.2 of 1 is perfect for most refraction effects. The first image in Figure 11.13 is raytraced with a refractive index of 1.2 on the wine bottle and glasses; the second image has a refractive index of 0.8 on both bottle and glasses.

Figure 11.13

Refractive index of 1.2 (left render); refractive index of 0.8 (middle render); setting the refraction quality in the Render Settings and the shader (right two windows)



You can specify whether an object is visible in a refracting object by clicking or clearing the Visible In Refractions check box in the Render Stats section of the object's Attribute Editor.

When rendering refractions, make sure the Refractions attribute under Raytracing Quality in the Maya Software tab of the Render Settings is set to at least 2 or higher; otherwise, your refraction may not appear properly. For mental ray, it is the Refractions attribute found under the Raytracing heading on the Quality tab, as shown in Figure 11.13 (right). You will also need to set the Refraction Limit found under the Raytrace Options heading in the Attribute Editor for the refractive shader in question.

Using Cameras

Cameras capture all the animation fun in the scene. The more you know about photography, the easier these concepts are to understand.

The term *camera*, in essence, refers to any perspective view. You can have as many cameras in the scene as you want, but it's wise to have a camera you're planning to render with placed to frame the shot and a different camera acting as the perspective work view so you can move around your scene as you work. The original persp view panel fits that latter role well, although it can be used as a render camera just as easily.

You can also render any of your work windows to test-render orthogonal views of your model the same way you render a perspective view.

Creating a Camera

The simplest way to create a new camera is to choose Panels → Perspective → New, as you've seen in a previous exercise. This creates a new camera node in Maya and sets that active panel to its view. Cameras will work in any renderer, though some renderers such as VRay may provide their own specialized cameras or camera attributes.

You can select a camera in a view panel and transform it (move it, rotate it, scale it) just as you would select and transform any other object in Maya to be animated or positioned. Furthermore, you can move a camera and rotate it using the Alt/Option key and mouse button combinations.

For example, click inside a new Maya Scene Perspective window to make it active. Select that view's camera by choosing View - Select Camera. The camera's attributes appear in the Channel Box. Try moving the view around using the Alt/Option key and mouse button combinations. Notice how the attributes change to reflect the new position and rotation of the camera. You can animate the camera—for example, zoom in or out or pan across the scene—by setting keyframes on any of these attributes.

Camera Types

You can create three types of nonstereo cameras for your scene: Camera; Camera And Aim; and Camera, Aim, And Up (also known as single-node, two-node, and three-node cameras, respectively). To create any of these cameras, choose Create → Cameras. You can also change the type of these cameras at any time through the Attribute Editor. The other two options for creating cameras are Stereo Camera and Multi Stereo Rig to allow for a stereoscopic effect, although they aren't covered in this book.

The single-node camera (Camera) is the most common (see Figure 11.14). This camera consists of a single camera node that you move and rotate as you would any other object for proper positioning. The persp view panel's camera is a single-node camera.

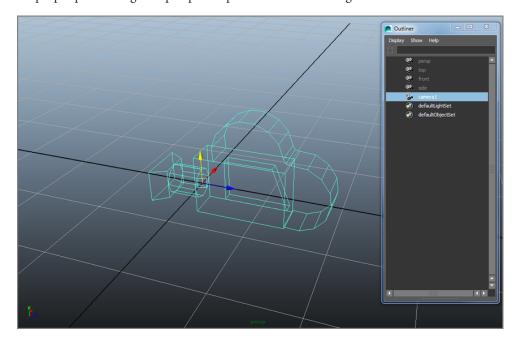


Figure 11.14
A single-node camera

The two-node camera (Camera And Aim) consists of the camera node and an aim node. You use the aim node to point the camera as opposed to rotating it to orient it properly. This is useful for animating a camera following an object. You animate the movement of the aim node to follow your object much as you'd follow a car around a racetrack. The camera pivots to follow its aim point and, hence, the object. (See Figure 11.15.)

The three-node camera (Camera, Aim, And Up) has a camera node, an aim node, and an up node. The additional up node is to orient the camera's up direction. This gives you the ability to animate the side-to-side rotation of the camera as well as its aim direction. (See Figure 11.16.)

Figure 11.15

A two-node camera

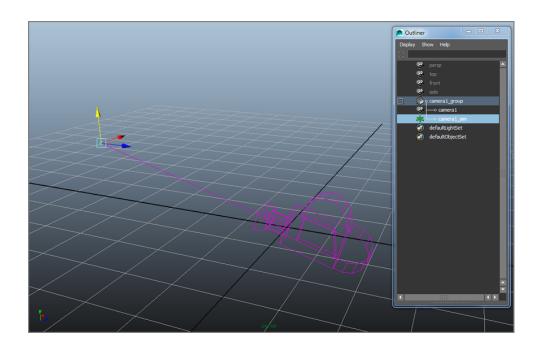
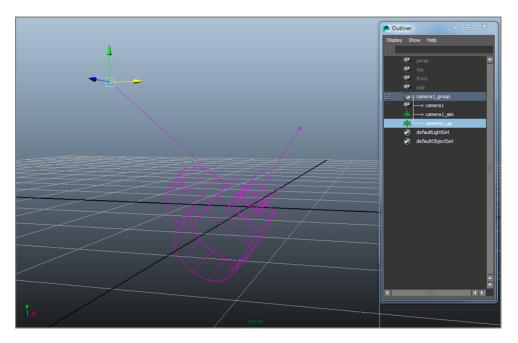


Figure 11.16

A three-node camera



As an example, download the still_life_render_v02.mb scene from the Lighting project on the book's web page, www.sybex.com/go/introducingmaya2015, and set your project to Lighting. You'll see a green box in the persp view panel that displays the resolution (set to 640×480) and the name of the camera (cameral).

Special attributes control the function of camera nodes. To set these attributes, follow these steps:

- 1. With cameral selected, open the Attribute Editor (press Ctrl+A or choose View → Camera Attribute Editor).
- 2. At the top of the window, select the type of camera controls you want.

 The Controls attribute sets the type of camera from single- to two- to three-control nodes. Figure 11.17 shows the Attribute Editor for the cameral camera.

Focal Length

The Focal Length attribute specifies the length of the lens. The lower the focal length (a.k.a. short lens), the wider the view. At very low numbers, however, the image is distorted, as you can see in the comparison in Figure 11.18. The higher the focal length, the closer the subject seems to the camera.

Although adjusting the Focal Length attribute of a camera zooms in and out, it isn't the same as moving the camera closer to your subject using the Alt+RMB+click procedure to zoom in view panels (which is called *trucking* the camera). Focal-length zooming can create optical distortions, such as can be created with a fish-eye lens.

If you need to match some CG element in Maya to a photograph or video that you've imported as an image plane,

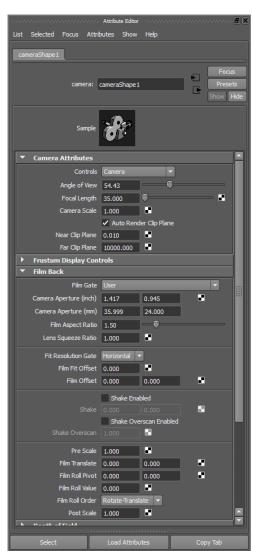
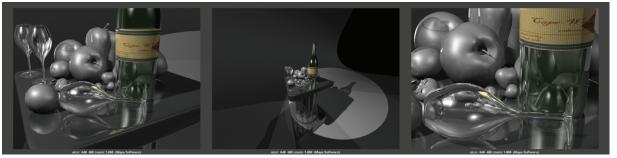


Figure 11.17

The Attribute Editor for the persp camera set your camera's focal length to match that of the real camera used for the background photo.



Focal Length = 35 Focal Length = 8 Focal Length = 60

Clipping Planes

All cameras in Maya have clipping planes that restrict the amount of information that can be seen through them. The clipping plane is defined by the Near Clip Plane and Far Clip Plane attributes. These set the minimum and maximum distances, respectively, of the clipping plane. Any object or portion of an object that passes beyond these distances won't show in the window and should not render.

If you notice objects disappearing as you move your camera and create a scene, it may be because of the clipping plane. Increase the Far Clip Plane attribute, and the objects should reappear in the view. You can display the near and far clipping planes of any camera with the check boxes in the Frustrum Display Controls section, as well as view your angle of view by turning on Display Frustrum.

Film Back

The Film Back attributes concern the type of output you'll be dealing with after your renders are finished and you're ready to put your animation on tape, DVD, film, or what have you.

Film Gate Defines the aspect ratio of your camera's view. Images that are output to HD television have an aspect ratio of 1:1.78, which funnily enough does not have a preset in the Film Gate drop-down list. For an HD camera output, simply set Film Aspect Ratio to **1.78**. For standard-definition TV output, simply select 35mm TV Projection from the Film Gate drop-down list. (For more on aspect ratios, see Chapter 1.)

Fit Resolution Gate Allows you to align footage you may have imported as an image plane to match up CG properly to live action.

Overscan

Found in the Display Options section for the camera's Attribute Editor, Overscan lets you resize the view without changing the film gate that will render. For example, the scene on the left in Figure 11.19 is set up with an Overscan setting of 1.3, allowing you to see more than what will render, which is defined by the outline box. The scene on the right in Figure 11.19 is set up with an Overscan setting of 2, which increases even more how much you see in the cameral panel but doesn't change the view when rendered.

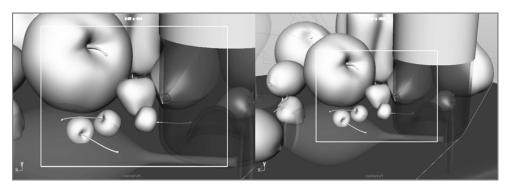


Figure 11.19

Overscan settings define how much you can see of your scene in the camera but not how much renders in the image.

You can turn the green box in the panel on and off through the camera's Attribute Editor. Also in the Display Options section are Display Film Gate and Display Resolution check boxes, shown in Figure 11.20. Ideally, these two green boxes should align perfectly in the view pane. If the resolution box (the solid green line) doesn't line up with the film gate box (the dashed green line), change your film gate selection to match the resolution's aspect

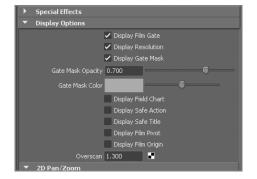


Figure 11.20
Camera display options

ratio in the Render Settings window. A resolution of 640×480 , for example, has an aspect ratio of 1.33, the same as the 35mm TV Projection film gate currently selected for this scene.

Environment

In the Environment section, you'll find attributes to adjust the background color that renders and to create an image plane, as shown in Figure 11.21.

If you want to use a solid color as the camera's background when you render, click the color swatch next to Background Color to change the background color in your



Figure 11.21

Adjusting the camera's environment and creating camera image planes

renders using the Color Chooser. The slider allows you to control the value, or brightness, of the current color. Neither changes the background color of your view panels, however.

CAMERA IMAGE PLANES

A camera image plane isn't like the reference planes you used for modeling the red wagon in Chapter 6, "Practical Experience." In this case, an image plane is created to be a background specific for that particular camera or view panel, but it's typically also used as a reference, much like the planes you created and mapped in Chapter 6. Camera image planes are useful when you're matching your scene to existing footage or an image. For example, if you need to animate a flying saucer into a home video of a family gathering, you would import the video as an image sequence into Maya through a perspective camera to be able to line up your UFO properly to zap your cousins.

You can import an image plane by clicking the Create button in the Environment section of the Attribute Editor (see Figure 11.21) or directly through a view panel's menu, as you'll see in the next exercise.

In this exercise, you'll learn how to import a sketch of an axe into the Front view panel for a modeling assignment. (You won't actually model the axe, however.) The image, a sketch of a simple axe design, is to be used as a template for outlining the model. You can find the file Axe_outline_1.tif in the sourceimages folder of the Axe project on the companion web page; it's shown in Figure 11.22.

Follow these steps:

- 1. Choose File → New Scene.
- 2. Import the sketch of the axe into Maya as a camera image plane for the Front view panel. In your Front window, choose View → Image Plane → Import Image.
- 3. Point to Axe_outline_1.tif in the Axe project's sourceimages folder and load it. The sketch displays in your Front window and as a selected plane in the Perspective window (see Figure 11.23). You should see the image plane's attributes in the Attribute Editor. At first, the plane looks a little small in your view panels, but you can adjust the size of the image plane in the Attribute Editor under the Placement Extras heading using the Width and Height attributes. Just make sure that the Maintain Pic Aspect Ratio box is checked. In Figure 11.23, the Width and Height settings have been adjusted to 25.0×17.325.

Now you'd be ready to trace the outline of the axe easily in the Front view panel, if you were going to model the axe.

If you can't see the image plane, click Show in the view panel and make sure Image Planes is checked.

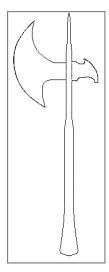


Figure 11.22 An image to import as a camera image plane

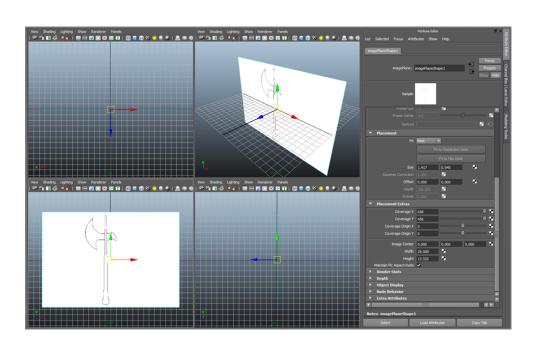


Figure 11.23

Importing a camera image plane into the Front view panel

IMAGE PLANE SEQUENCE

A movie file or a sequence of files can also be brought in to animate or to track motion (a.k.a. *matchmoving*) as a camera image plane. It's generally best to use a frame sequence, however. When you bring in an image for an image plane, check the Use Image Sequence box in the image plane's Attribute Editor window, as shown in Figure 11.24. Maya will automatically load the image to correspond to the frame number in the scene. For example, at frame 29 in your Maya animation, Maya loads frame 29 of your image sequence. But your image file sequence must be numbered correctly (such as *filename.###.tif*). You can import an image plane into any perspective view in the same way.

If the clutter of seeing a camera image plane in the other windows bothers you, under Image Plane Attributes in the Attribute Editor, change the radio button selection next to Display from In All Views to Looking Through Camera. This setting removes the image plane from the other windows.

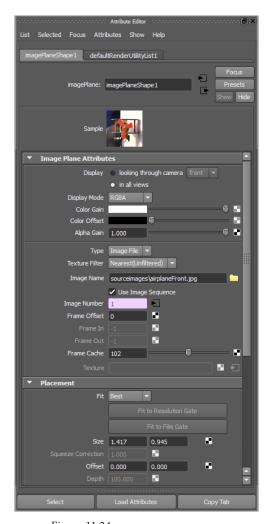


Figure 11.24 Importing a sequence of image files as a camera image plane

Motion Blur

Motion blur is an optical phenomenon that occurs when an object moves quickly in front of a camera: the object looks blurred as it crosses the frame. Maya Software rendering renders motion blur in two ways—2D blur or 3D blur—although neither will render as reflections.

- In the 2D blur process, Maya calculates after the frame is rendered. Any objects moving in the frame are blurred with a 2D filter effect. The 2D blur is effective for most applications and faster than 3D blur.
- The 3D blur process is calculated while a frame of the sequence is rendering. Every motion blur-enabled object is blurred with typically better results than 2D blur but at a cost of a much longer render time.

I'll briefly cover motion blur in mental ray for Maya later in the chapter.

To enable motion blur for the Maya Software renderer, open the Render Settings window. In the Motion Blur section on the Maya Software tab, click the Motion Blur check box. Then, choose 2D or 3D blur.

Typically, you control the amount of blur rendered for 2D and 3D by setting the Blur By Frame attribute—the higher the number, the greater the blur. Using additional controls, however, you can increase or decrease the 2D blur effect in the render. The Blur Length attribute affects the streakiness of the blur to further increase or decrease the amount of motion blur set with the Blur By Frame attribute.

Setting a camera's Shutter Angle attribute (in the camera's Attribute Editor in the Special Effects section) also affects the amount of blur rendered—the higher the number, the greater the blur.

Batch Rendering

So far, you've used single-frame rendering numerous times to see a scene in the Render View window. But how do you start rendering an animation sequence to disk? This is called batch rendering in Maya, whichever renderer you use. To batch-render an entire scene, follow these steps:

1. Open the Render Settings window.

2. Choose Maya Software to use to render, enter the start and end frames of your animation, and select your image format. Select your quality and resolution settings. Finally, set the camera you want to render in the Renderable Camera attribute.

Be sure to select *name*. #. ext in the Frame/Animation Ext drop-down list to render a sequence of files. Remember, if you leave the default setting, which is *name*. ext, only a single frame renders.

- 3. In the main Maya window under the Rendering menu set, choose Render ¬ Batch Render □ to open the Batch Render Animation dialog box (or the Batch Render Frame box when rendering a single image). Figure 11.25 shows the Maya Software rendering batch options on top, and the bottom image shows the batch options for mental ray rendering.
- 4. If you have a multiprocessor, hyperthreading, or dual-core machine, select how many CPUs you would like to use to render your scene. A value of 0 will use all processors on the machine.
- 5. Click Batch Render to render the frame range you specified in the Render Settings window. The render occurs in the background, and you see progress updates in the Command line at the bottom of your Maya screen and in the Script Editor window if you open it.

To see a frame as the batch render progresses, choose Render → Show Batch Render.
To cancel a batch render, choose Render →
Cancel Batch Render.

When you batch render, your image files are written to the images folder of the current project. Make sure your project is properly set; otherwise, your files will end up in an unexpected folder. You can always see the render path and the image name at the top of the Render Settings window, as shown in Figure 11.26.

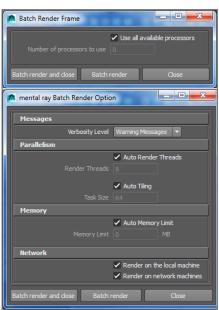
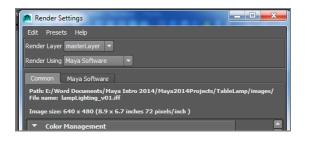


Figure 11.25

The Batch Render Frame dialog box for Maya Software (top) and mental ray rendering (bottom)

Figure 11.26

The Render Settings window shows you where the images will be rendered.



Rendering the Wine Bottle

In this section, you'll set up and render an animated camera to move over 25 frames of a wine bottle still life using Maya Software rendering at first.

Set your current project to the Lighting project downloaded from the book's web page and then load still_life_render_v01.mb. You'll adjust your render settings and some shader properties to make the wine bottle look more like glass.

Selecting Render Settings Options

Set your resolution and quality settings in the Render Settings window.

- 1. Open the Render Settings window and select Maya Software. Click the Maya Software tab.
- 2. From the Quality drop-down list, select Production Quality. Doing so presets the appropriate settings to produce a high-quality render.
- 3. Click the Common tab. Set Frame/Animation Ext to name.#.ext, set Start Frame to 1, set End Frame to 25, and set Frame Padding to 2.
- 4. From the Image Format drop-down list, select TIFF.
- 5. Make sure Renderable Camera is set to cameral. In the Image Size section, set Presets to 640×480 .

Setting Up the Scene

Now, set up some of the objects in the scene. The wine bottle has been imported into the still life scene, and three wine glasses have been added. All the lights are in place, as is the camera.

Start by setting up this scene to raytrace to get true reflections and refractions.

- 1. Turn on refractions for the Glass shaders. In the Hypershade, select the Glasses material and open the Attribute Editor. In the Raytrace Options section, click the Refractions check box, and set Refractive Index to 1.2. Set Reflection Limit to 2. Select the Wine_Bottle material and repeat the previous steps.
- 2. You need to change your lights' shadows to raytraced shadows. Remember that semitransparent objects cast solid shadows unless shadows are raytraced, so the glasses and wine bottle will cast shadows as if they were solid and not glass. In the Outliner, select spotLight1. In the Attribute Editor, in the Shadows section, enable Use Ray Trace Shadows. Repeat these steps for the remaining two shadow-casting lights: spotLight4 and spotLight3. Figure 11.27 shows the three shadow-casting lights in the scene in white.
- 3. Open the Render Settings window and turn on raytracing in the Raytracing Quality section. Set Reflections to 2.

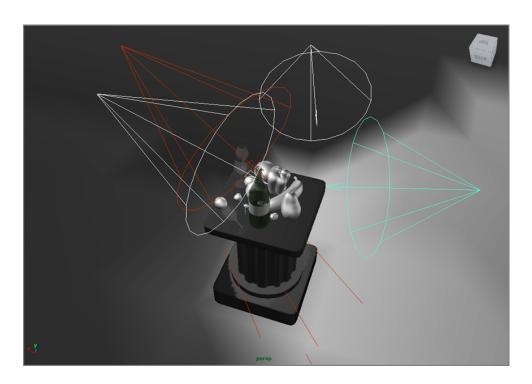


Figure 11.27 Shadow lights

You can't select all three lights at once to turn on raytraced shadows in the Attribute Editor. Any adjustments you make in the Attribute Editor affect only the most recently selected object, not multiple selections.

Setting Up the Camera

Next, you'll set up the camera to render the scene.

- 1. Make sure you are looking through the cameral camera. Open the camera's Attribute Editor through the cameral panel (choose View → Camera Attribute Editor).
- 2. Select the Display Film Gate option in the Display Options section to turn on a dashed green box in the cameral panel. Enable the Display Resolution option. Notice that the two boxes aren't aligned.
- 3. Because the resolution is 640×480, you'll use a 1.33 aspect ratio. Select 35mm TV Projection from the Film Gate drop-down list. The two green boxes now align. Although it's not absolutely necessary to match the resolution with the film gate, it's definitely good practice to do so, especially if you'll later insert CG in live-action videos.

- 4. As soon as you change the film gate, the framing of the scene changes. You must move the camera out to frame the entire still life. If you try to use the Alt key and mouse button combinations to zoom out, you'll notice that you can't move the camera in this scene: The movement attributes for the camera have been locked to prevent accidental movement that would disrupt the shot.
- 5. To unlock the camera, choose View Select Camera. The camera's attributes appear in the Channel Box. Some are grayed out, signifying that they're locked and can't be changed. Highlight the locked attributes and RMB+click Unlock Selected in the Channel Box.
- 6. Create an animated camera move to pull out and reveal the still life slowly over 25 frames. Set your Range slider to 1 to 25 frames. Go to frame 1. The cameral view should be similar to that shown in Figure 11.28. Select all three Translate and Rotate channels in the Channel Box and RMB+click Key Selected to set keyframes for the first camera position.
- 7. Zoom (actually, truck) the camera out by pressing Alt+RMB+click to a wider framing to reveal the entire still life. Highlight the Translate and Rotate channels again in the Channel Box and set keyframes for them. (See Figure 11.29.) Scrub your animation, and you'll see a pullout revealing the full scene.

You can lock the camera to prevent accidentally moving the view after you set your keyframes, especially if Auto Keyframe is on. Select the Translate and Rotate attributes in the Channel Box, RMB+click and choose Lock Selected.

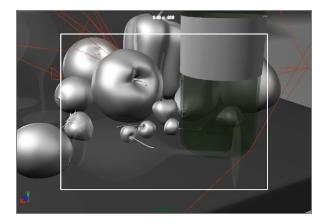


Figure 11.28 The camera view at the beginning of the animation

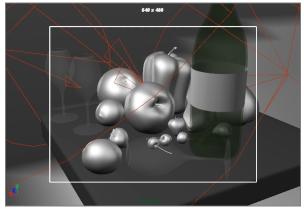


Figure 11.29 Pull out the camera.

Batch Rendering and Playing Back the Sequence

Now, you're ready to render the 25-frame sequence. Choose Render → Batch Render.

Because you're raytracing this scene at full resolution, this render could take 20 minutes or longer. To chart the progress of the render, open the Script Editor by clicking its

icon on the Help line or by choosing Window → General Editors → Script Editor.

To see the frames play back, you'll need a program that can load the images in sequence and play them back for you. You can also import the image sequence into a compositing or editing program, such as Adobe After Effects or Premiere, to play back as a clip and edit as you like.

You can also use FCheck, a frame viewer that is included with Maya. This small and surprisingly powerful program plays back your images in real time so that you can judge your finished animation. To use FCheck, follow these steps:

- 1. In Windows, choose Start → All Programs → Autodesk → Autodesk Maya 2014 → FCheck to open the FCheck window, as shown in Figure 11.30.
- 2. Choose File → Open Animation.
- 3. In the file browser, find your images folder in your project and click the first frame of the sequence you want to play back. FCheck loads the images frame by frame into RAM and then plays them back in real time. Just set your playback speed and use the DVD controls to play back your sequence.

mental ray for Maya

You had some experience with mental ray rendering in the previous chapter as you lighted using Physical Sun and Sky, which automatically used Final Gather to render. In this part of the chapter, I'll discuss mental ray options to begin to scratch the surface of this powerful renderer.

First, if you haven't done so already, be sure that mental ray is loaded. When you first start up, mental ray for Maya may not load because it's considered a plug-in. Choose Window → Settings/Preferences → Plug-in Manager to open the Plug-in Manager, shown in Figure 11.31. Make sure both the Loaded and Auto Load check boxes for Mayatomr.mll (or Mayatomr.bundle for Mac users) are checked so that mental ray for Maya loads by default.

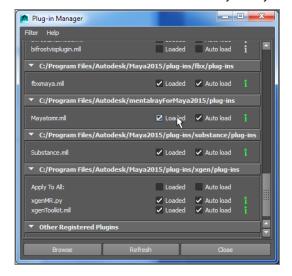
Figure 11.30

FCheck, shown here with a sample image, plays back your rendered sequence.



Figure 11.31

Use the Plug-in Manager to load mental ray for Maya.



To render with mental ray, open the Render Settings window and change the Render Using drop-down menu selection to mental ray, as shown in Figure 11.32 (shown with the Features tab selected). The Render Settings window now has the Common tab along with five other mental ray-specific tabs: Passes, Features, Quality, Indirect Lighting, and Options.

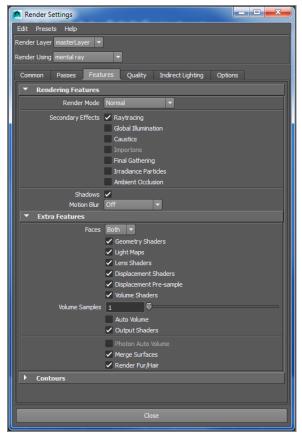


Figure 11.32

The Render Settings window for mental ray, showing the Features tab

mental ray Quality Settings

As with the Maya Software renderer, the primary quality settings for the mental ray renderer center on antialiasing. However, mental ray for Maya offers you finer control over how you set the quality levels through the Render Settings window. On the Quality tab, you'll find these attributes:

Sampling Mode (Under the Sampling Heading) This drop-down menu sets the type of sampling mental ray performs in the render. Sampling is the number of times mental ray reads and compares the color values of adjacent pixels in order to smooth the resulting render to avoid jagged lines. The choices are Unified Sampling (default), Legacy Rasterizer Mode, and Legacy Sampling Mode. Unified Sampling is generally the best choice because it sets the best range based on the Min and Max Samples you set along with a single Quality value. With the Min and Max Samples set to 1 and 100, respectively, the Quality value slider is key. A Quality setting at about 1.0 is considered good for production-quality renders, though you may need slightly higher than 1.0 in some cases. Quality of about 0.2 is a good fast render for previews. Experimenting with Min and Max Sample values will make more minor adjustments to the look of your

render than would making changes to Quality.

For those who are already familiar with previous mental ray workflows, Legacy Sampling Mode will give you access to additional sampling modes available in previous versions of mental ray for Maya. Legacy Sampling is discussed in the "Legacy Sampling Mode" sidebar. However, Unified Sampling mode attempts to greatly simplify the quality settings for mental ray for Maya and is preferred for use.

Sample Options Heading Filtering is done on the results of the sampling of pixels to blend the pixels of a region together to form a coherent image. A high filter size tends to blur the image, whereas low filter values may look overly crisp. Gauss is the default filter and gives a slightly softer result to help deal with any grainy image noise. Usually, render times don't vary much between different filter modes—it's not as if Gaussian takes four times longer to render the same frame over the Box filter type. But each gives a slightly different type of softness depending on its Filter Size setting.

LEGACY SAMPLING MODE

When the Sampling Mode option is set to Legacy Sampling, you will have access to more controls over the quality of mental ray renders, which may be helpful if you are familiar with previous versions of mental ray. Under Sampling Mode, you will see the Legacy Sampling Mode drop-down menu: Fixed Sampling, Adaptive Sampling (default), and Custom Sampling. Adaptive Sampling is generally the best choice because it sets the best range based on the Max Sample Level option you set. If you choose Custom, you can set your own sampling range with the Min and Max Sample Level options.

Number of Samples These Min and Max Sample Level values set the number of times mental ray samples a pixel to determine how best to anti-alias the result. The higher the number, the finer the detail and the smoother the appearance of your rendered lines will be. These settings are exponential, so a small increase yields a much greater quality and much longer render time. You won't need to set Max Sample Level above 2 for most uses. Both Max and Min Samples Level can be set into the negative numbers for fast renders, although you'll notice jagged edges. A good setting for most renders is 2 for Max Sample Level with the Adaptive Sampling mode.

What determines the exact sample level within the min and max range set by these attributes depends on the Anti-aliasing Contrast Threshold value.

Anti-aliasing Contrast The values here determine when mental ray turns up the number of samples in a particular region of the frame. If the contrast level from one pixel to its neighbor is below the threshold value, mental ray turns up the number of samples for that pixel to render a higher-quality result. Therefore, the lower the values set here, the higher the sample rate will be (within the Min and Max levels set).

Sample Lock is turned on to reduce noise and artifacts in rendered sequences that have lots of movement.

Rather than setting high sample rates with just the Min and Max Sample Level attributes, carefully increase the Quality value to force mental ray to use a sample rate closer to the Max Samples only where it needs to do so. You'll see this in action in the next section.

When you're ready to render your scene to disk, you still use batch rendering; however, the options in the Batch Render Option window are different (Figure 11.33) than they are for Maya Software.

Figure 11.33 mental ray batch render options

Render Settings in Action!

In this section, you'll look at how Quality values and how the number of Min and Max Samples work together to determine the quality of the render of a toy wagon. You can find the scene (RedWagonRenderSettings.ma) to render in the RedWagon project on the book's web page. Make sure to set your project to RedWagon.

Open the Render Settings window, making sure you are set to mental ray, and open the Quality tab. Set your Sampling Mode value to Unified Sampling. Occasionally you may see that Sampling Mode reverts to Legacy Sampling Mode when you open a file even though it was set to Unified Sampling. If that is the case, any settings you've created for Unified Sampling Mode do remain intact. Quality should be set to **0.50**, Min Samples to **1.0**, and Max Samples

to 1.0 in this file already.

These render quality settings are low at first and will yield the render shown in Figure 11.34, which took just less than two minutes. You can see jagged highlights on the wagon, especially the white lines on the side body, the reflections in the front black nose, and the edge of the white rims on the back wheel.

If you increase Max Samples to **100** and Min Samples to **1.0**, you'll see an immediate increase in quality (especially in the reflections of the wagon and the white decal lines) and a noticeable increase in render time (about 50 percent as long), as shown in Figure 11.35, which took just under three minutes to render.



Figure 11.34

A toy wagon rendered with low sampling values. Notice the jagged highlights on the white decal lines and the back wheel.



Figure 11.35

Better sampling improves the wagon's appearance.

You can get an even cleaner render. If you crank up the Quality value to the heavens, such as 2.0, you get a far cleaner render than before but with a much longer render time. Figure 11.36 shows the same frame with the Quality value set to 2.0 instead of the default of 0.5. The Min Samples is still set to 1, and the Max Samples is left at 100. The wood panels and the reflections of the wagon in the checkered floor are both noticeably nicer in this render. Your results as you render the scene on your own computer will be more noticeable than the images shown here. But this render took 4:35.

For a very quick render to check composition and lighting, set your Quality, Min, and Max Samples values all to **0.1**. You should get a super-fast render, though good luck discerning much detail. It's easy enough to crank the numbers to the sky, but render times will quickly become unacceptable, especially when you have a supervisor or client breathing down your neck. Find just the right combination of numbers for acceptable results in a minimum of time. Each scene will be different.

Motion Blur with mental ray

Now that you have a primer on quality settings in mental ray for Maya, you'll learn how motion blur works.

 Load the Planets_motionBlur.ma scene file from the Solar_System project on the web page. This is an animated scene of the solar system with just the first few planets and moons, as shown in Figure 11.37. A blast from the past!

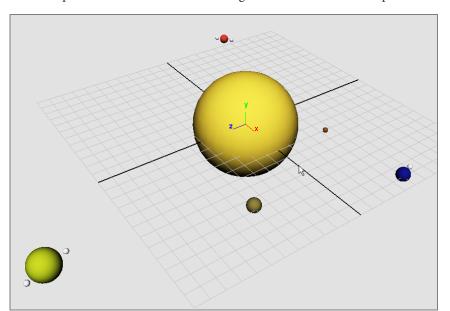


Figure 11.36

The wagon has still better sampling with a higher Quality value. You'll see it better in your own renders.

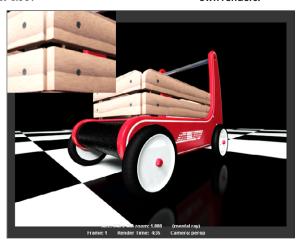


Figure 11.37
The Solar_System project is back!

- 2. Open the Render Settings window. The scene should already be set to render with mental ray. If your scene isn't, make sure the mental ray for Maya plug-in is loaded and set the scene to render with mental ray.
- 3. On the Common tab, set Frame/Animation Ext to *name.#.ext*, Frame Padding to 3, Start Frame to 1, and End Frame to 100.
- 4. Set Image Size to 640×480 for now and leave the renderable camera to persp.
- 5. Go to frame 75 and render a frame from the persp view panel. Figure 11.38 shows the result.
- 6. In Render Settings, click to open the Features tab. In the Rendering Features heading, select No Deformation in the Motion Blur drop-down menu, as shown in Figure 11.39. This setting is the faster of two motion blur methods and works for objects that move in the scene without deformation. *Deformations* occur when the mesh or surface of an object changes, such as when you use a lattice or a skeleton rig like IK to drive a character. For deforming objects, you need to use Full as the Motion Blur setting.
- 7. Click to open the Quality tab in Render Settings. Set Sampling Mode to Unified Sampling, and set Quality to **0.5** and Min Samples and Max Samples each to **1.0**. In the Motion Blur → Motion Blur Optimization heading, set the Motion Blur By attribute from 1.0 to **3.0** for quite a bit of blur. Render a frame at frame 75. (See Figure 11.40.)

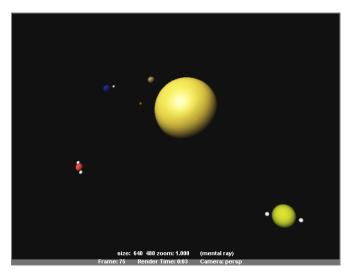


Figure 11.38

A render of frame 75 shows the planets in motion without any motion blur.

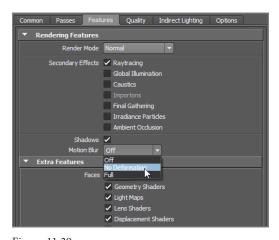


Figure 11.39
Turn on motion blur.

8. The render is very fast and shows quite a bit of blur; however, the quality is low. You see a lot of graininess in your render, especially with the yellow and red planets at the bottom of the screen. On the Quality tab of the Render Settings window, leave Sampling Mode at Unified Sampling, but change Max Samples to **50**. This render should be fast but cleaner, with a smoother but still-grainy motion blur for the planets. Notice, however, that the render time increases by about 50 percent! Now try Max Samples at **100** and Quality at **2.0** (Figure 11.41 shows this much cleaner render). You control the amount of motion blur in your scene with the Motion Blur By attribute (step 7) for your scene.

Render Layers

Most of the time, it's best to composite different elements together to form a final CG image. Professional CG workflow almost always requires multiple render passes that are composited together later for the maximum in efficiency and quality.

Maya does a great job of making rendering in layers much easier with render layers. As you saw earlier in this book, using display layers helps a lot in keeping your scene organized. Render layers operate in basically the same way, although they function by separating different elements of the scene into separate renders.

I'll address the most basic and commonly utilized render layer functionality here: separating objects into different ren-

ders. You'll select elements in a scene and assign them to different render layers. When you batch-render the scene, Maya will render each of the layers separately and save the

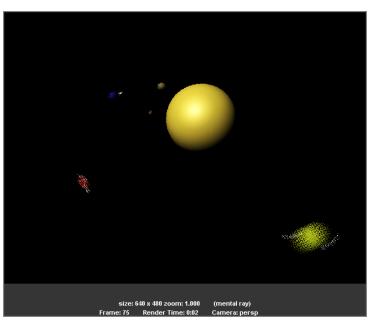


Figure 11.40
Render a frame with motion blur turned on.

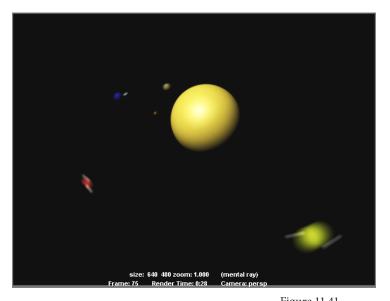


Figure 11.41
Motion blur looks much better with higher sampling levels.

files into their own subfolders in the images folder of your current project. You'll then need to load all the different rendered layers into a compositing program, such as The Foundry's NUKE or Adobe After Effects, and composite the layers together.

Render Passes in mental ray

Two powerful features in mental ray rendering, Render Passes and Pass Contribution Maps, make rendering in passes and elements much more efficient. Passes allow you to separate shadows, reflections, diffuse color, and so on, from the render to give the most control in compositing over the image. This mental ray-specific workflow is fairly advanced and requires an existing knowledge of rendering and compositing render passes (such as you will do with render layers) to fully grasp. Thus, I won't cover these features in this book; however, you should be aware of this mental ray rendering pipeline as you move beyond this introduction and continue rendering with mental ray in your own work.

Rendering the Still Life in Layers

In this example, you'll separate a scene into different layers for rendering with mental ray, though the render layer workflow here applies to Maya Software rendering as well.

To separate a scene into different layers, follow these steps:

- 1. Set your project to the Lighting project downloaded from the book's web page and open the still life scene (still_life_renderLayers_v01.mb from the scenes folder to start there). The lights and a camera are already set up in this scene. Set your camera view to cameral. Open Render Settings and make sure mental ray is the current renderer. Set Sampling Mode to Unified Sampling, set Quality to 2.0, and set Min/Max Samples to **1.0** and **100**, respectively.
- 2. To separate the scene into different renders, in the Layer Editor, click the Render tab to switch to the Render-Layers view. Select the foreground lemon and click the Create New Layer And Assign Selected Objects icon (). Doing so creates a new render layer called layer1 and assigns the lemon to it. (See Figure 11.42.)
- 3. Click layer1, and everything but the lemon disappears from the scene (see Figure 11.42). Double-click layer1 in the Layer Editor and rename it **lemonPass**, as shown in Figure 11.43.
- 4. If you test-render a frame, the frame should turn black. You need to assign your lights to the render layer as well. Select all the lights in the scene using the Outliner, as shown in Figure 11.44. Right-click the lemonPass layer in the Layer Editor and select Add Selected Objects from the context menu. If you render a test frame now, the lemon renders as shown in Figure 11.45.

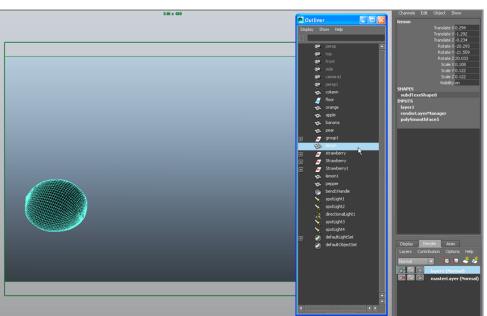


Figure 11.42

The newly created render layer with the foreground lemon assigned



Figure 11.43

Rename the render layer to lemonPass.

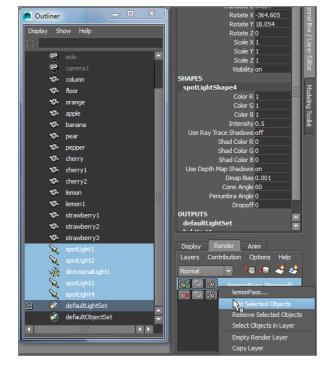
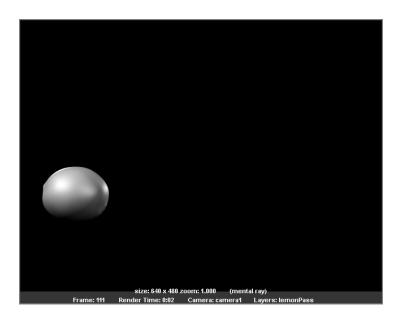


Figure 11.44

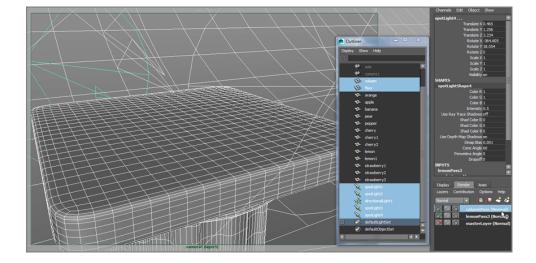
Add the lights to the lemonPass render layer.

Figure 11.45 The lemon rendered on its very own layer



5. Create a render layer for the column and floor. Click the masterLayer render layer and then select the column, the floor, and all the lights in the scene. As you did with the lemon, click the Create New Layer And Assign Selected Objects icon () and then click the new layer to select and display it. Double-click the layer name to rename the render layer columnPass (see Figure 11.46).

Figure 11.46 The background is assigned to its own render layer.



6. Click back to the masterLayer; then select all the rest of the pieces of fruit, as shown in Figure 11.47, and create their render layer called fruitPass. You should now have

three render layers and the default masterLayer. The masterLayer is always present to house all the elements of the scene. It switches to being not renderable as soon as you create a new render layer, as evidenced by the little red *X* in its Renderable icon (**S**).

Figure 11.47 The newly created fruitPass render

layer

- 7. Using the first icon on each of the render layer entries in the Layer Editor, you can toggle whether that layer will render. Make sure all your render layers are renderable, and leave the masterLayer off. Because all the elements of the scene are assigned to one render layer or another, the whole scene is covered. The masterLayer nonetheless is always present because it represents all the objects in the scene, assigned to layers or not.
- 8. With the fruitPass render layer selected, test-render a frame, as shown in Figure 11.48. You should see just the fruit and not the background, column, or foreground lemon. Notice the reflections in the fruit.

Test-Rendering Everything Together

By default, Maya renders only the selected visible render layer. You can, however, tell Maya to test-render and show you all the layers composited together to give you a preview of what you'll end up with when you composite all the layers together after batch-rendering the scene.

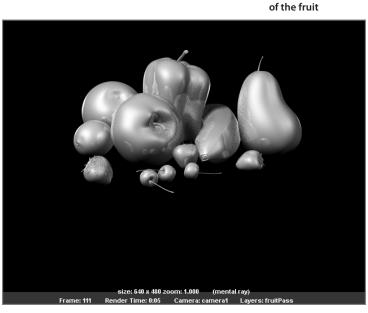


Figure 11.48 Rendering the rest

To test-render all the layers together, click the Options menu in the Layer Editor, as shown in Figure 11.49, and toggle on Render All Layers by clicking the check box if it isn't already selected.

Figure 11.49 Turn on Render All Layers.



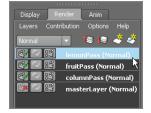
Now if you test-render a frame, Maya will render each layer separately and then composite them together. Test-render a frame with Render All Layers enabled, and you should notice that the foreground lemon is missing, as in Figure 11.50.

Figure 11.50 Rendering all the layers together



This is because the render layers in the Layer Editor need to be reordered so that columnPass is on the bottom, fruitPass is in the middle, and lemonPass is on top. This is the layer ordering for the composite. You can reorder the render layers by MMB+clicking

Figure 11.51 Reorder the render layers.



a layer and dragging it up or down to its new location in the Layer Editor, as in Figure 11.51.

Now if you render the frame with Render All Layers enabled, you see the scene properly placed, as shown in Figure 11.52. Notice that shadows and reflections on the column's tabletop are missing, however. For that, you will need

to create render layer overrides to allow reflections and shadows of an object to render, but without the object itself rendering.

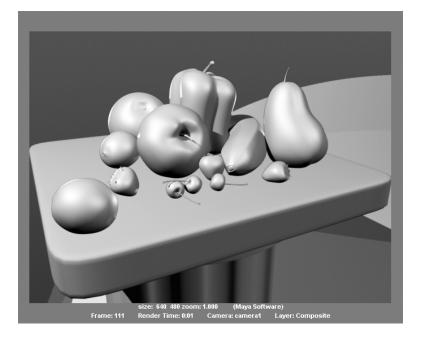


Figure 11.52

Now the lemon appears in the render.

Render Layer Overrides

Where are the shadows and reflections? If you recall from earlier in this chapter, this scene renders with shadows, and this scene should have reflections as well. However, the reflections and shadow-casting objects are on different layers than the column. Shadows and reflections can't be cast from one layer onto another. In the following steps, you'll use render layer overrides to set up your columnPass layer to have shadows and reflections from the fruit but not render the fruit on top of the column:

Render layers and overrides are powerful workflow tools in Maya. Because there can be an obscene number of overrides in any given render layer, it's important to follow through this workflow carefully. Practice it to gain a comfort level before getting too fancy with render layers and overrides. Also, it helps to keep notes beside your keyboard to help you stay organized with render layers and overrides. And lastly, there are occasional problems when assigning shaders to the faces of a mesh (as opposed to the entire mesh) when using render layers. Try to always assign shaders to the entire mesh and not to faces in those cases. If you need multiple shaders on a single mesh, instead use Layered shaders and UV maps.

Figure 11.53

Add the fruit back into the laver. 1. Click to the columnPass render layer. In the Outliner, select all the fruit objects and add them to the layer (Figure 11.53).

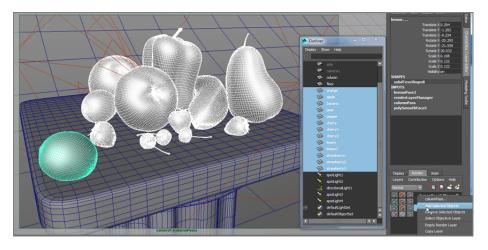


Figure 11.54 **Turn off Primary**

Visibility to set a render layer override for the orange on the columnPass laver.



- 2. If you render the layer now, it will render the fruit. To turn them off without removing them from the layer (and hence lose shadows and reflections), you'll set render layer overrides for each of the fruit objects. Select the orange, and open the Attribute Editor.
- 3. On the orangeShape tab in the Attribute Editor, click to open the Render Stats heading. Here you see switches allowing for different ways to render the object. You want

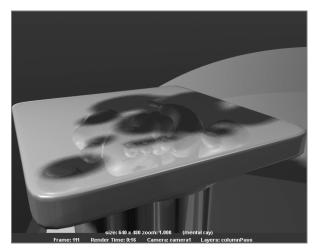
the orange to cast reflections and shadows in the scene, without rendering itself, so uncheck the Primary Visibility box. It will turn orange to signify that there is now a render layer override on that attribute for the current render layer (columnPass) only (Figure 11.54).

As a matter of fact, if you click the other render layers with the orange still selected, you will notice that the Primary Visibility setting for the orange turns back on and loses its orange color in the Attribute Editor on the other render layers. Render layer overrides work only on the layers in which they were created.

4. Now select each fruit model down the list in the Outliner individually and turn off Primary Visibility one by one. You cannot select all the objects at once and turn off Primary Visibility in the Attribute Editor; it will register the action only for the last selected object and not them all.

To change attribute values on multiple objects at once, such as Primary Visibility, you can use Window → General Editors → Attribute Spread Sheet. Be careful, though!

- 5. Turn off Render All Layers (in the Layer Editor, uncheck Options Render All Layers). Render the columnPass layer, and you'll see the top of the column is rendering with shadows and reflections but not the fruit! (See Figure 11.55.)
- 6. Finally, you make the shadows look nicer by using raytrace shadows. Select spot-Light1, and in the Attribute Editor, under the Shadows → Depth Map Shadow Attributes heading, turn off Use Depth Map Shadows. Scroll down a bit until you see the Raytrace Shadow Attributes heading; open it, and turn on Use Ray Trace Shadows.
- 7. Repeat step 6 for spotLight3 and spotLight4. Render, and you'll see sharp shadows now on the column (Figure 11.56).





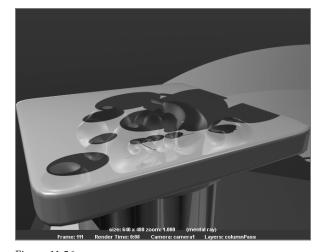
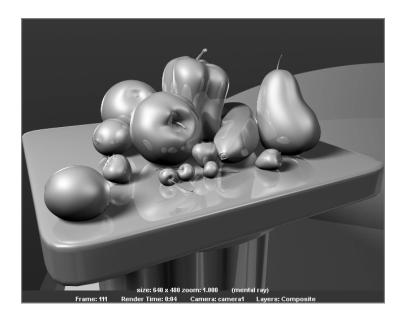


Figure 11.56
The raytraced shadows are clean but too sharp.

- 8. Let's enable soft shadows now for those three spotlights. Select spotLight1, and in the Raytrace Shadow Attributes, set Light Radius to **0.4** and Shadow Rays to **16**. Repeat the procedure for spotLight3 and spotLight4. The shadows will look much more believable now.
- 9. Turn on Render All Layers, and render a frame to see the entire composition (Figure 11.57). Once you verify everything is working and looking good, you can now render your scene into separate layers for better control in compositing.

Figure 11.57

The layers all rendered together with soft shadows and reflections



You can check your work against the file still_life_renderLayers_v02.mb from the scenes folder of the Lighting project on the book's web page.

Batch Rendering with Render Layers

Now that you have everything separated into render layers, let's talk about batch-rendering the scene to composite later. Make your selected render layers renderable by making sure the Renderable icon next to each render layer is enabled, as shown in Figure 11.58.

Figure 11.58 Check to make sure the lavers you want to render are enabled.



In the Render Settings window, make sure Renderable Camera is set to cameral, as shown in Figure 11.59. Select your image file type and anything else you need to set in the Render Settings window. Then, in the main Maya window, choose Render - Batch Render to render the scene. Because render layers are enabled here, Maya renders the scene into different subfolders under the images folder of your current project. Each render layer gets its own folder in the images folder, as shown in Figure 11.60.

I have only begun to scratch the surface here. When you get the hang of rendering and as your CG needs begin to grow, you'll find a plethora of options when rendering with layers and rendering with passes through mental ray. This section and the "Ambient Occlusion" section later in this chapter are here only to familiarize you with the basic workflow of render layers. You can find a wealth of information about rendering in layers and passes in the Maya online documentation under the Help menu.

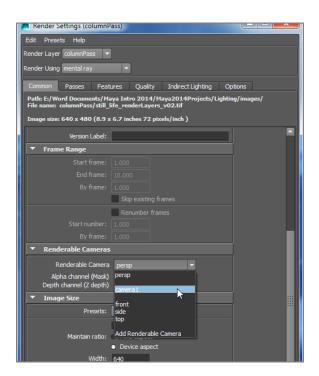


Figure 11.59

Select camera1 as your renderable camera in the Render Settings window.

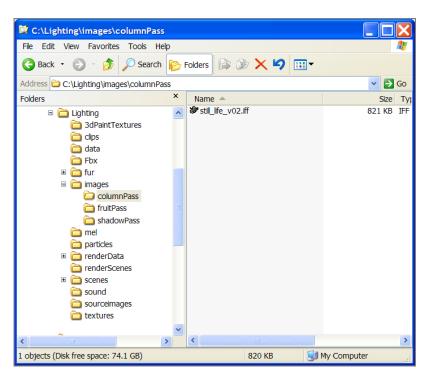


Figure 11.60

Render layers batch-render into subfolders in the images folder of your project.

Final Gather

Final Gather (FG) is a type of rendering with mental ray that includes a simple light bounce within the scene. Final Gather traces light as it reflects off surfaces to illuminate the scene for a nicely realistic render that takes into account color bleed of light from one surface to another. For example, a red wall casts a red hue on the surface right next to it. FG is an intricate dance of settings and numbers that lets you get perfect renders. It's tough to cover in an introductory book such as this; however, this section will give you a primer to start using FG. In a follow-up section, you'll use a combination of lights and HDRI with Final Gather to render the decorative box.

The basic premise is that FG uses the illumination in the scene from lights as well as objects with color and incandescence set in their shaders to create a soft natural light. Its base settings will give you nice results right from the start.

To render the still life fruit scene with FG, begin by creating a dome light, which evenly illuminates a scene from a dome around the scene. (The term *dome light* is a bit of a misnomer, because a dome light isn't itself a light.) You'll construct a simple NURBS sphere, cut it in half to create a dome, and give it an Incandescent shader to provide the light for the scene. This type of quick FG setup is extremely useful for rendering out soft lighting and shadows to show off a model or a composition.

To light and render with FG, follow these steps:

- 1. Ensure that mental ray for Maya is loaded.
- 2. Load the still life scene file still_life_mentalray_v01.mb from the Lighting project from the book's web page. In the persp view panel (not the cameral panel), create a polygonal sphere, and scale it up to enclose the entire scene. Select the bottom half faces of the sphere, delete them to cut the sphere in half, and name the object light-**Dome**. Place the dome to fit over the scene, as shown in Figure 11.61.
- 3. Add an Incandescent shader to the dome to give the scene some illumination through FG. To do so, in the Hypershade create a new Lambert shader. Turn up its incandescence to a medium light gray. Assign the new shader to the dome.
- 4. Maya automatically creates a general default light in a scene that has no lights when you try to render it. This is so you can test-render your scene quickly. However, because this incandescent dome light should be the only light source in the FG render, you have to turn off the default light feature before you render. Open the Render Settings window and, under the Common tab's Render Options heading (at the bottom of the Attribute Editor), turn off the Enable Default Light check box.

Don't forget to turn Enable Default Light back on when you're finished with this FG exercise.

Figure 11.61

Create a hemisphere to act as a light dome.

- 5. At the top of the Render Settings window, switch the Render Using attribute to mental ray. You can keep the settings on the Common tab to render at 640×480. Choose the Quality tab to access its settings. Set Sampling Mode to Unified Sampling, and leave Min and Max Samples at 1.0 and 100, respectively, but set a Quality value of **0.3**. Check to make sure that raytracing is enabled under the Raytracing heading. Figure 11.62 shows the Quality tab of the Render Settings window.
- 6. Click to open the Indirect Lighting tab and click the Final Gathering heading to expand its attributes. Turn on Final Gathering by checking the box. You can leave the settings at their defaults for your first render. Figure 11.63 shows the Indirect Lighting tab for the Render Settings window with Final Gathering turned on.
- 7. You need to make sure the sphere you're using as a dome light doesn't render out in the scene. Select the sphere and, in its Attribute Editor under the Render Stats heading, turn off Primary Visibility. Since there are no render layers in the scene, it doesn't turn orange like with the previous section's exercise. Highlight the cameral panel, and render the frame. Maya makes two passes at the scene and shows you something like Figure 11.64. You can control the brightness of the scene with the Incandescence attribute value of the dome's shader—the whiter the incandescence, the brighter the scene.

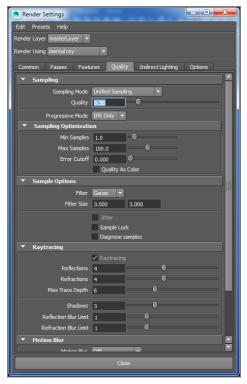
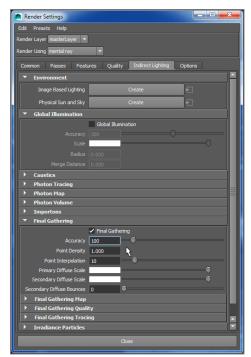


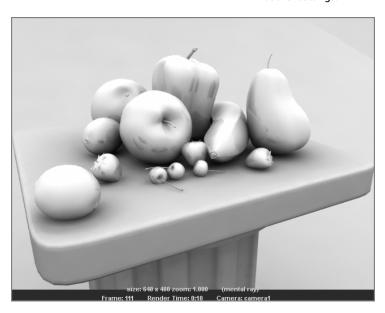
Figure 11.62 The Quality tab



The Indirect Lighting tab accesses all the Final Gather settings.

Figure 11.63

Figure 11.64 The Final Gather render of the still life



You can adjust the level of lighting by increasing or decreasing the amount of incandescence on the light dome's shader. The proximity of the sphere also affects the light amount, so moving the sphere closer or farther away will change the lighting level as well. You can also insert lights into the scene as you see fit.

You can improve the quality of the render by adjusting any of the following three settings on the Indirect Lighting tab under the Final Gathering section of the Render Settings window:

Accuracy Increase this number for a more accurate render. Don't increase it too much, however, because it will slow down your render. Earlier the still life was rendered with the default Accuracy setting of 100. In Figure 11.65, which uses a setting of 500, the soft shadows are noticeably smoother. Accuracy dictates the number of FG rays that are cast into the scene. The more rays, the better the quality of the render, although at a cost of rendering time.

Primary Diffuse Scale Use this setting to adjust the overall lighting level globally in the scene. The higher the value of this color, the brighter the scene. You can type values greater than 1 in the Color Chooser.

Secondary Diffuse Bounces Setting this attribute to 1 will add an additional bounce of light inside your scene, essentially brightening everything. Shadowed areas will pick up the biggest boost, and the scene lighting will be a bit softer overall.

The Point Density and Point Interpolation attributes, along with several other attributes in the Final Gathering section, optimize and further control the FG settings for optimum efficiency size: 840 x 480 zoom: 1.000 (mental ray)

Figure 11.65

Accuracy set at 500 increases the smoothness of the soft shadows. It's hard to see here, but you'll notice it in your own render.

and results, especially with animated scenes. Consider using the basic settings shown in this chapter to become familiar with mental ray for Maya, and then use FG to crank out fast and elegant renders to show off your models until you get more comfortable with mental ray.

Final Gather is a tough nut to crack; it will take you some time to become proficient at rendering with FG in scene files and especially with animated scenes. Later in this chapter, you'll try your hand at using FG, HDRI, and regular lights to render the decorative box from the previous chapters.

Ambient Occlusion

Ambient Occlusion is a special render pass that helps add depth and reality to a render. Ambient Occlusion goes by the premise that when two objects or surfaces are close to each other, they reduce the amount of light at the intersection. Ambient Occlusion passes make for great contact shadows and bring out the definition in surface creases and corners very nicely. Figure 11.66 shows an Ambient Occlusion pass for the living room scene from the PDF Global Illumination exercise included on the book's web page.

Figure 11.66 The Ambient Occlusion pass is black and white and is used to darken areas of the original color render.



How the composite works is simple. Ambient Occlusion gives you a black-and-white pass of the same geometry you've already rendered. This pass is then multiplied over the color render. That means a brightness value of white (a value of 1) in the Ambient Occlusion pass won't change the color of the original render (when the original color of the render is multiplied by 1, it stays the same color). The black areas of the ambient occlusion image (with a brightness value of 0) turn the same areas of the original render black (when the original color of the render is multiplied by 0, it goes to black). The gray points of the multiplying image darken the original render. It sounds confusing, but when you see it, it makes much better sense.

The Living Room

You'll now take an existing render of a living room and add an Ambient Occlusion pass using render layers. Set your current project to the Livingroom project you copied from the web page, open the livingRoom_v2.mb file from the scenes folder, and follow these steps:

1. Make sure mental ray is loaded, of course, and that Render Settings is set to Render Using mental ray. As you did earlier, you need to create a new render layer for the Ambient Occlusion. This layer requires all the objects in the scene but the lights to be assigned to it. In this scene, the only light is a single Directional light. In the Outliner, select all the top nodes of the scene, but leave out the light, as shown in Figure 11.67.

- 2. Click the Render tab in the Layer Editor to switch to Render and then click the Create New Layer And Assign Selected Objects icon (3). Doing so creates a new layer (layer1) along with the preexisting master-Layer, as shown in Figure 11.68.
- 3. Click the new layer (layer1) to activate it. Everything in the scene should display as it did before, although the light disappears from view. Double-click layer1 and rename it ambientOcclusion.
- 4. You're going to use a preset to create a material override. This takes all the objects in the scene and assigns a single material to them, in this case an Ambient Occlusion shader that generates the Ambient Occlusion pass for the entire scene. RMB+click the ambientOcclusion layer, and select Attributes. In the Attribute Editor, click and hold down the Presets but-





Figure 11.67 Select the scene objects for your Ambient Occlusion pass.

Figure 11.68 Create a new render layer.

ton. From the menu that appears, select Occlusion, as shown in Figure 11.69.

If you're in Shaded mode, everything should turn black in the view panels. This is normal, because everything now has the Ambient Occlusion Surface shader assigned, and that shader displays black in Maya panels. If you click the masterLayer, everything pops back into place.

5. When you select Occlusion from the Presets menu, your Attribute Editor window should display a new shader called surfaceShader1. Figure 11.70 shows the new



Figure 11.69 Setting an Occlusion preset for the layer

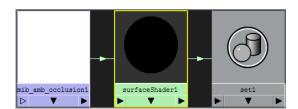


Figure 11.70

The Ambient Occlusion shader in the Hypershade

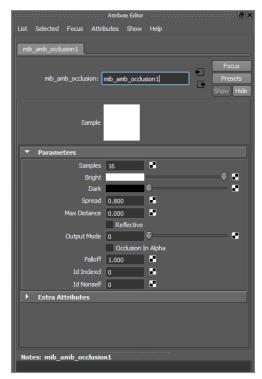
- Ambient Occlusion shader that is assigned throughout the scene. Rename that shader to ambOccShader. Don't worry if you don't see the surfaceShader1 in your Attribute Editor; you'll come back to it in later steps.
- 6. While still in the ambientOcclusion layer, render a frame. You should see something like Figure 11.71. You needn't worry about most of the settings in the Render Settings window; the layer preset takes care of it all. Well, almost all. It doesn't turn off the Global Illumination pass that you used for the original color render in the PDF exercise (which you can find on the web page); you need to turn that off manually in the next step.

Figure 11.71

The first render from the Ambient Occlusion pass doesn't look greatit's too dark.

- 7. In the Render Settings window, on the Indirect Lighting tab, you need to turn off Global Illumination, but just for the ambientOcclusion layer. With ambientOcclusion still selected in the Layer Editor, RMB+click the Global Illumination check box in Render Settings and choose Create Layer Override. The attribute lettering turns orange to signify that it's being overridden for that render layer, just like you saw before. Uncheck the box. It doesn't turn off GI for the masterLayer pass. You don't have to spend the time calculating any GI for this pass.
- 8. The render is mostly black, which darkens the original color render to black almost everywhere. You need the Ambient Occlusion shader to render mostly white with some darkening at the corners and where objects contact each other. You can adjust the Ambient Occlusion shader to fix this. Open the Hypershade window and click ambOcc-Shader (or surfaceShader1, if you didn't rename it earlier in step 5—rename it now).
- 9. Notice that the Out Color attribute has a texture connection to it (signified by the icon). Click this button to display the mib_amb_occlusion1 texture node in the Attribute Editor, as shown in Figure 11.72.

10. Set the Max Distance attribute to **4.0**, as shown in Figure 11.73, and render the frame again. Your Ambient Occlusion layer should look like Figure 11.74. It should have also taken less than half as long to render as the darker render from step 6.



Samples 16
Bright

Dark
Spread 0.800
Max Distance 4.000
Reflective
Output Mode 0
Occlusion In Alpha
Falloff 1.000
Id Indexcd 0
Id Nonself 0
Fatra Attributes

Figure 11.72
The Ambient Occlusion shader attributes

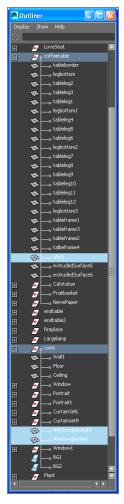
Figure 11.73
Set Max Distance to 4.



Figure 11.74
The Ambient
Occlusion layer pass
looks much better,
but you aren't finished yet!

Figure 11.75

The Outliner view of the glass geometry in the scene



- 11. Notice that the glass in the window and the glass on the coffee table have shadows on them. Because glass is clear, they shouldn't have any Ambient Occlusion applied to them; it would look odd in the final composite. Select those pieces of geometry (Figure 11.75 shows the Outliner view for those pieces).
- 12. With the glass geometry selected, select Display → Hide → Hide Selection. Now the glass won't render in the Ambient Occlusion pass. But because you hid the objects only in this render layer, they still appear in the masterLayer; they will still render as glass in the color pass.
- 13. Render the frame, and you should have something similar to Figure 11.76. This is the Ambient Occlusion pass you need to composite.

Remember to check out the PDF exercise on the web page if you'd like to learn how this scene was lit with Global Illumination.

Rendering the Results

You could save the image you just rendered in the Render View window to use in the composite and then render the masterLayer for the color pass in the Render View and save that frame as well. Instead, let's batch-render the scene to show how Maya handles rendering with layers enabled. To batch-render the scene, follow these steps:

- 1. Turn on the Renderable icon box () for the masterLayer, and make sure it's on for the ambientOcclusion layer as well.
- 2. Click to activate the masterLayer. Open the Render Settings window, and on the Common tab, verify all the settings to render a single frame (name.ext.# in the



Figure 11.76

The Ambient Occlusion pass

- Frame/Animation Ext field) at 640×360 (which is half 720p HD resolution). Also, select cameral as the renderable camera.
- 3. If you batch-render (by choosing Render → Batch Render), Maya renders both render layers into the images folder of your project in separate folders, as shown in Figure 11.77.



Figure 11.77 Maya renders the layers into their own folders by default.

If for some reason your renders don't show up in your project's images folder, open the Script Editor window and look at the batch-render report. The render feedback shows you where the rendered images were saved, as shown in Figure 11.78. You may also want to make sure you set your project to the Livingroom project from the web page.

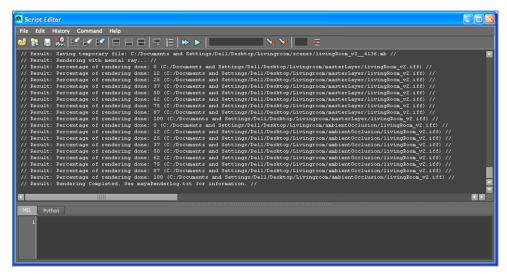


Figure 11.78

The Script Editor gives you feedback on the progress of the batch render.

Compositing the Results

Now that the two layers are rendered in their respective folders, load them into your favorite compositing package. You'll layer the Ambient Occlusion pass over the color render using a Multiply Transfer mode. This exercise uses Adobe After Effects CS4 to demonstrate how the Ambient Occlusion pass is composited over the original color render.

Figure 11.79 shows After Effects with the masterLayer color pass loaded. Figure 11.80 shows the ambientOcclusion pass layered on top of the color layer. Finally, Figure 11.81 shows the ambientOcclusion pass changed to a Multiply Transfer mode (as it's called in After Effects). Notice how the dark areas of the Ambient Occlusion pass help give contact shadows and depth to the color pass. Voilà!

This is a prime example of rendering different passes to achieve a more realistic result. Use different layers to put your final images together in composite.

You can see the difference Ambient Occlusion made to the living-room image in the Color Section of this book and on the web page.



Figure 11.79

The masterLayer render pass is loaded into After Effects.

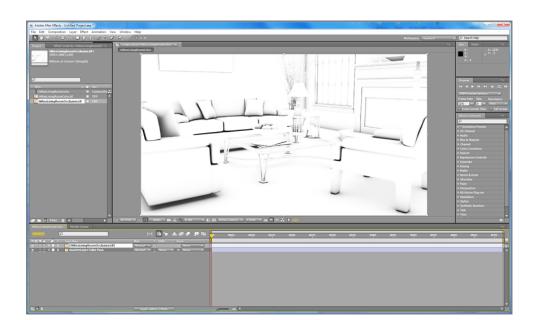


Figure 11.80

The ambientOcclusion render pass is loaded into After Effects.



Figure 11.81

The ambientOcclusion pass is multiplied over the color pass and creates a more realistic image.

HDRI

As you saw in the Final Gather section, FG rendering is based on the illumination in the scene from lights as well as the brightness of objects in the scene, such as a light dome. In a previous section, you used an incandescent dome to light the still-life scene. But what if you were to use an image instead of just white for the light dome?

Furthermore, what if the image you used was a high dynamic range image (HDRI)? Several photos at varying exposures are taken of the same subject, ranging from very dark (low exposure), which highlight only the brightest parts of the scene, and going all the way up to very bright (overexposure), which capture the absolute darkest parts of the scene. When these images (usually five or seven images) are compiled into an HDR image, you get a fantastic range of bright to dark for that one subject.

How does this help you light? With image-based lighting (IBL), mental ray creates an environment sphere in your scene to which you assign an image, usually an HDRI. That environment sphere, much like the white dome in the Final Gather exercise earlier in the chapter, uses the brightness of its image to cast light in your Maya scene.

The best type of image to capture for an IBL is sometimes called a *light probe*. This is a picture of an environment, such as the office reflected in a chrome ball shown on the left of Figure 11.82 or the stitched-together panoramic stills compiled into a map shown on the right of Figure 11.82. The stitched panoramic on the right of Figure 11.82 was photographed at four 90-degree angles using a fish-eye camera lens capable of capturing a field of view of close to 180 degrees using a panoramic tripod attachment.

Figure 11.82 A light-probe photo of a desk is taken with a chrome ball (left) and a panoramic stitch (right).



Figure 11.82 shows the middle exposure of five exposures taken of the office and one of the four angles of the living room. Figure 11.83 shows five images in the range from underexposed (dark) to overexposed (bright) photos that were used to compile the HDRIs shown here. You will use an HDRI made from the living-room panoramic photos to light the decorative box in the next section.



Figure 11.83

Five exposures make up the HDRI of a desk using a chrome ball (left); and a living room using a panoramic stitch (right), which is to be used in the next section.

The living-room photos are taken at 90-degree intervals inside the living room and then stitched and compiled in an HDRI file using a photographic software package (shown in Figure 11.84). The HDRI file is called livingRoomPanoramic.hdr and is in the sourceimages folder of the TableLamp project on the web page for the next section's IBL lighting exercise. You won't be able to see the extensive range of the HDR because the great majority of computer displays are limited to a display of 8-bit color.

Figure 11.84

Fish-eye photos are taken at four angles of the living room, each with several exposures from light to dark (above). The fisheye photos are then warped and stitched together using the handy software package PTGui and merged into an HDR image file (bottom).



The HDR will be mapped onto an IBL sphere. This is a large sphere that surrounds the environment in your Maya scene. The individual photos first need to be stitched together and converted to a rectangular image file, as shown in Figure 11.84. The full spherical panorama space captured in these photos and stitched together is laid out into a rectangular format that is suitable to project onto a sphere in Maya, just as if a geographic map on a school room globe was unwrapped into a rectangular sheet of paper.

Software such as HDRShop, available online for free, allows you to combine multiple images into an HDR. One useful program, PTGui from the Netherlands (available for a demo online at www.ptgui.com), allows you to stitch panoramas as well as merge HDR images, which makes it a fantastic tool for CG lighters and photographers. As long as the photos to be stitched were taken with enough overlap at the edges, PTGui is able to turn them into a large panoramic photo. And if the proper exposure ranges were also taken, PTGui creates a solid HDR as well. I won't get into the details of creating the HDRI because it's an advanced topic using Adobe Photoshop, PTGui, and/or other software; however, it's good to know the origins of HDR images and how they come to be used in an IBL.

Displacement Mapping the Decorative Box

Before continuing with the lighting and rendering for the lamp and decorative box, let's first address a need for detailing just the box by itself. Look at the photo of the box in Figure 3.21 (in Chapter 3, "The Autodesk" Maya* 2015 Interface") and compare it to the renders of the box in Chapter 10 (Figure 10.65); you'll see that the carved details in the CG box need more detail. Let's start by taking a closer look at the box by itself.

- 1. Make sure your project is set to the Plane project. Open the file planeLightingScene_ v05.mb from the Plane project. An error may come up saying that the reference file cannot be found (if so, it will look like Figure 11.85). If that is the case, simply click Browse and navigate to where you have saved the Decorative_Box project's scenes folder, where you will find the scene file boxDetailO1.mb. Otherwise, the scene should open with no issues. There is a copy of the boxDetailO1.mb file in the scenes folder for the plane project as well.
 - Oh no! Reference file not found! Reference file not found

Reference file not found: file:///E:/Word Documents/Maya Intro 2013/Maya2013Projects/Decorative_Box/scenes/boxLighting01.mb

- 2. Select the top node of the toy plane (planeGroup) and create a new display layer called **planeLayer**. Turn off visibility for the planeLayer.
- 3. In the Render Settings window, change the renderer to mental ray.
- 4. Use the persp camera (not the renderCam) and line up a view of the box to match the box render shown in Figure 11.86.

Make path changes permanent

Remember these settings

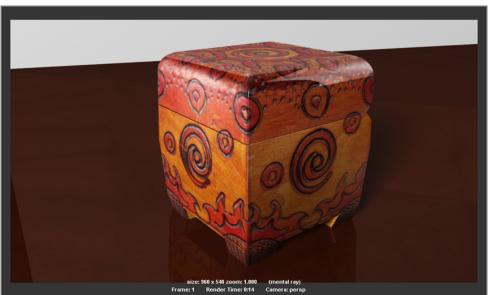


Figure 11.86

Figure 11.85

The decorative box's current look when you finished the lighting in Chapter 10

- 5. Render a frame, and you should notice more closely the lack of definition in the box when compared to the real photo of the box (Figure 3.21 in Chapter 3).
- 6. Save your work!

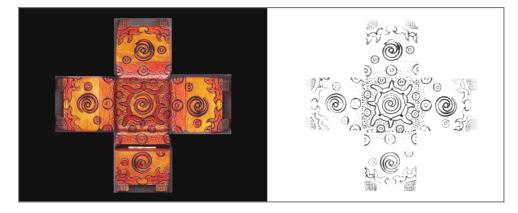
In the current render, the grooves are flat and shiny, whereas in the real box, they're carved into the wood and aren't glossy like the rest of the box. You will add definition to the model through textures next.

You can use the scene file planeLightingScene_v06.ma to catch up to this point or compare your camera angles.

Reflection Map

Just as you use a map to put color on the box, you can use similar maps to specify where and how much the box will reflect. Figure 11.87 shows the color map you used to texture the box in Chapter 7 side-by-side with a black-and-white map that shows only the carved regions of the box. This map was created with elbow grease and hard work to manually isolate just the carved areas of the box in Photoshop. See how much I do for you?

Figure 11.87 The black-and-white map on the right shows only the carved areas of the box.



You'll first take the reflections out of the carved areas in the following steps:

- 1. Set your project to the Decorative_Box project from the companion website.
- 2. Open the boxDetail02.mb scene. This scene has the box set up with a few lights, a render camera and render settings preset, and a ground plane for you to focus on detailing just the box. Render a frame and study the outcome. Save it to the image buffer in the Render View window.
- 3. Open the Hypershade. In the Create bar on the left, click 2D Textures under the Maya heading and choose File to open a new image file node (called file1). See Figure 11.88.



Figure 11.88 Create a new file

node.

4. Double-click the file node and navigate to find and choose the boxCarvings.jpg file in the sourceimages folder of the Decorative_Box project. See Figure 11.89.

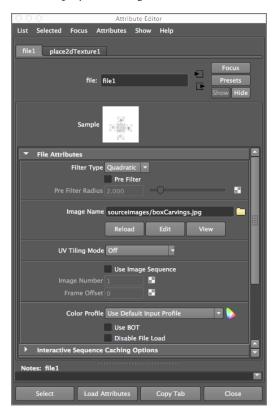


Figure 11.89 Select the proper image file.

5. If the sample swatch at the top of the Attribute Editor for the file1 node doesn't show the image you just selected, it is because the image is too large for Maya to create a thumbnail swatch automatically. If this is the case, the file node in the Hypershade would also be blank. RMB+click the file node and choose Refresh Swatch from the marking menu. Figure 11.90 shows the swatch with the black-and-white image as a thumbnail. This isn't necessary, but it makes it easier for you to identify what file map is where.

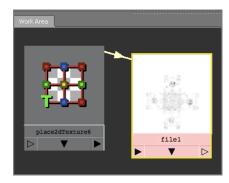


Figure 11.90

The swatch shows a thumbnail of the image you're using.

- 6. MMB+drag the phong1 node from the top of the Hypershade window down to the Work Area, alongside the file1 node.
- 7. Double-click the phong1 node to open the Attribute Editor. MMB+drag the file1 node onto the Reflectivity attribute for the Phong1 shader, as shown in Figure 11.91.
- 8. Notice the connection line in the Hypershade between the file1 and phong1 nodes. The white areas of the map tell the phong1 shader to have a reflectivity of 1, whereas the black areas have zero reflectivity. This effectively removes reflections from the box's carvings. Render a frame of the box in the renderCam view and compare it to Figure 11.92.

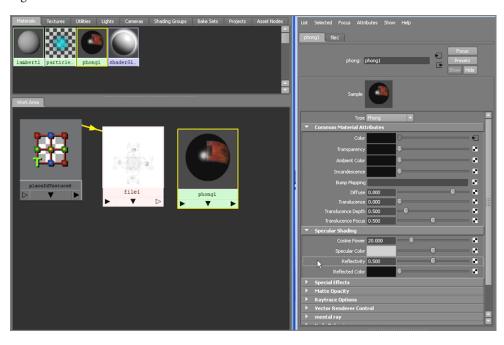


Figure 11.91 Map the new image file to the Reflectivity attribute.

- 9. In the Render View window, click the Keep Image icon () to store this render in the render buffer. Doing so allows you to easily compare renders to see the changes you make as you continue.
- 10. The box's reflections are now stronger all over, and the carvings still reflect. You need to turn on one switch. Select the file1 node and open the Color Balance heading in the Attribute Editor. Check the box for the Alpha Is Luminance attribute. This instructs Maya to use the luminance values (basically the brightness) of the image to output to the reflectivity attribute of the phong1 shader. The box renders properly, and the carvings have no reflections, as shown in Figure 11.93. Save this render into the render buffer with the Keep Image icon ().

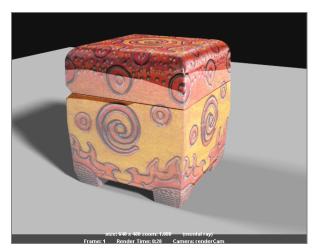
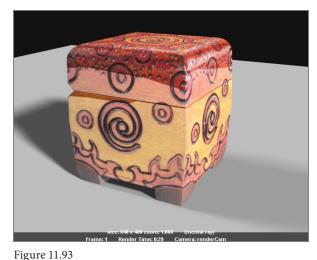


Figure 11.92 The reflections are stronger; the entire box looks like a mirror.



The carvings don't reflect, but the rest of the box's reflections are still too strong.

11. You need to reduce the reflections on the rest of the box. Double-click the file1 node to open the Attribute Editor. In the Color Balance heading, set Alpha Gain to **0.4**, as shown in Figure 11.94. This sets the brightest part of the reflections to be capped at 0.4 and not the previous 1.0. Render, and you see a much better reflection in the box and no reflections in the carvings. (See Figure 11.95.) Save this to the render buffer.

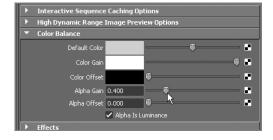
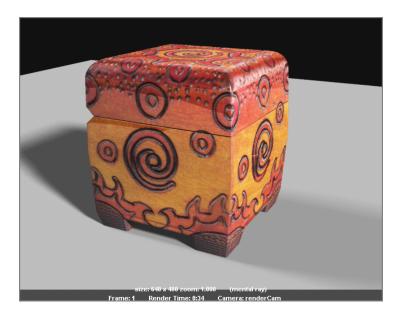


Figure 11.94 Set Alpha Gain to 0.4.

Figure 11.95 The reflections look much better now.



12. Rename the file1 node to **reflectionMap** and the phong1 node to **boxShader**, and save your work to a new version.

You can open the scene file boxDetail03.ma from the scenes folder in the Decorative_ Box project to catch up to this point or to check your work so far.

Displacement Mapping

Now that the reflections are set up, you'll use the same image map to create displacements in the box to sink the carvings into the box. Isn't that convenient?

- 1. In the Hypershade, click to create a new file node, and select the boxCarvings.jpg file in the sourceimages folder again. You don't want to reuse the same file node as you did for the reflections (reflectionMap node), because you need to change some of its attributes. Name the new file node **displacementMap**.
- 2. Double-click the boxShader node to open the Attribute Editor and MMB+drag the shader to the Work Area of the Hypershade. Then, MMB+drag the new displacementMap node onto the boxShader node in the Hypershade's Work Area and select Displacement Map from the context menu, as shown in Figure 11.96.
- 3. A new node appears called displacementShader1, but there is no connection apparent to the boxShader. Select the boxShader and click the Input And Output Connections icon at the top of the Hypershade (🔁). The Hypershade shows you the entire shader network, including the displacement map connection, as shown in Figure 11.97. Notice that the displacement connects to the phong ISG node and not directly to the boxShader node. This is normal and how it's always done.

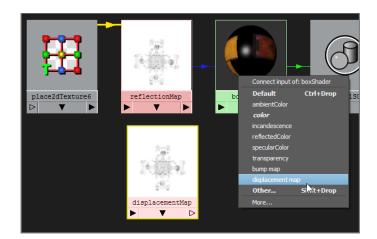


Figure 11.96

Add the new image map to the **Displacement Map** attribute.

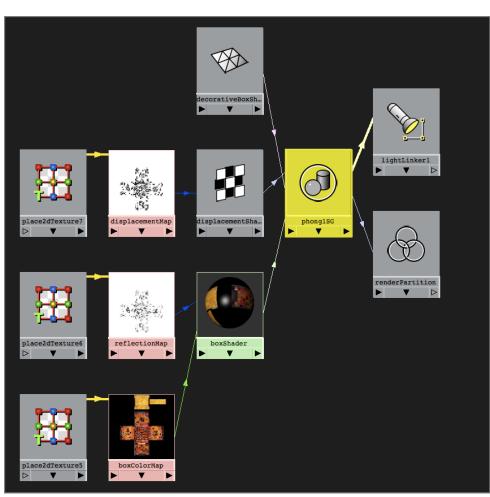


Figure 11.97 The shader network for the box

Figure 11.98

The displacements seem to be wrong! 4. Render the box, and see what happens now that you have a displacement map applied. Figure 11.98 shows how the box seems to have exploded!



Occasionally when you click to view the input and output of a shader as you did in step 3, the Hypershade will put some nodes on top of others. If you don't see any of the nodes, try clicking and dragging some of the nodes around in the Hypershade to see whether there are any accidentally hidden nodes. These little touches make Maya extra fun!

Not to worry—this is to be expected. The values coming from the black-and-white map dictate how much of the geometry is displaced (moved). Obviously, you don't have an image that works well for the amount of displacement, although it does look like the displacements are in the proper place to correspond to where the carvings are on the box.

5. Double-click the displacementMap node. In the Attribute Editor, open the Effects heading, and click the Invert check box. This turns the background to black and the carvings to white. Set Alpha Gain to 0.03 in the Color Balance heading, and make sure Alpha Is Luminance is also checked. (See Figure 11.99.) The Alpha Gain value reduces the amount of displacement, and inverting the image lets the rest of the box outside of the carvings not be displaced.

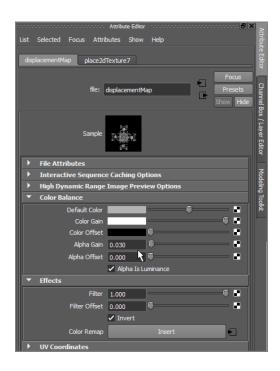


Figure 11.99 Set the attributes for the displacement image.

6. Render a frame, and compare it to Figure 11.100. It's not looking very good. The box isn't exploding anymore, and only the carvings are displacing, which is good. However, the carvings are sticking out, instead of in.

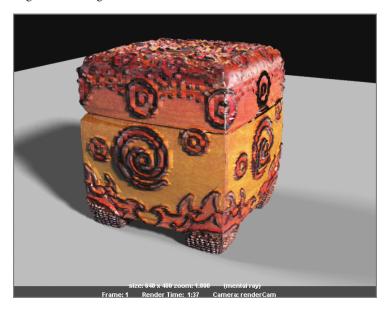


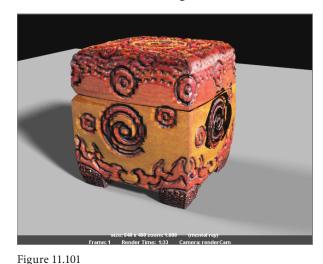
Figure 11.100 The carvings are displacing outward.

7. Set the Alpha Gain attribute to -0.015 to push the carvings in a little instead of out. Render and compare to Figure 11.101. The carvings are set into the wood as they should be, but the render looks bad. The displacements are very jagged and ill-defined.

USING SCALE ATTRIBUTE FOR THE DISPLACEMENTSHADER NODE **INSTEAD OF ALPHA GAIN**

In this exercise you are using the Alpha Gain attribute on the file node to control the amount of the displacement created by the carvings in the box. When you use mental ray, you sometimes could use the Scale attribute found in the dispalcementShader1 node created instead. For example, in this exercise, instead of setting Alpha Gain to -0.015 as in Step 7, leave Alpha Gain at 1.0 (default). Click on the displacementShader1 node and set the Scale attribute to -0.015 and render. Compare the two results, and they should be the same. This attribute doesn't work all the time, however, so getting used to using Alpha Gain to control the amount of displacement is the sure-fire way to do it.

8. The jagged look of the carvings has to do with the tessellation of the box—that is, the detail of the box's polygonal mesh. There are advanced ways to address how mental ray deals with displacements; however, the most basic method is to adjust the detail of the box's mesh. Figure 11.102 shows the current box's mesh. Notice that the sides of the box are simply detailed. To increase the box's faces to add detail, select the box, and then, in the Polygons menu set, choose Edit Mesh → Add Divisions □, under the FACE section. Set the options to Exponentially and Division Levels to 3, as shown in Figure 11.103. Click Add Divisions to execute.



The carvings are inset into the wood, but they don't look good yet.



Figure 11.102 The current box's mesh isn't detailed enough.

9. Figure 11.104 shows the newly tessellated box. It's far more detailed in the mesh. When you render, it gives a result similar to Figure 11.105. You can compare your work to the scene file boxDetail04.ma in the scenes folder of the Decorative Box project.

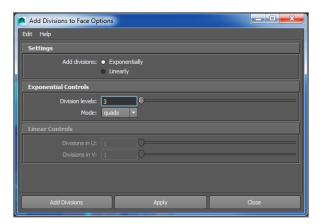


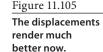
Figure 11.103 Set the Add Divisions values.

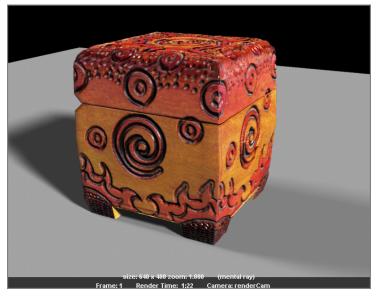
Figure 11.104 Added divisions

10. Now that you have the detail you want in the box, you need to prepare the box's scene file to be referenced back into your lamp lighting scene. Select the ground plane, the renderCam camera, and the three lights in the scene and delete them. Save your work to a new version.

Compare your work to the scene file boxDetail05.ma in the scenes folder of the Decorative Box project.

Notice that the render times increased because you increased the detail on the mesh in step 8. This increase is normal, but it can become a problem if the mesh is too divided and has more faces than you need. This box has fairly detailed carvings, so the level to which you set Add Divisions in step 8 gives a good result without being too detailed. As your experience increases in Maya and mental ray rendering, you'll learn how to use functions such as the mental ray Approximation Editor to keep the overall detail low on the mesh but with smooth displacement results when you





render. This is an advanced technique, so I won't cover it in this book. You've taken in quite a lot by this point already.

The next time you select the box and open the Attribute Editor, you'll see a new tab called polySubdFace1. This is the node added with the Add Divisions function in step 8. Feel free to adjust the Subdivision Levels setting on that tab and render to see how the detail levels affect the final render. Also try better quality settings in Render Settings, as you did with the red wagon earlier in this chapter, to clean up the render to your satisfaction.

You'll next add an IBL sphere and use the HDR image you saw earlier in the chapter.

Rendering the Scene with mental ray

You're all grown up and ready to light a full scene using Final Gather, IBL with an HDRI, and regular lights to get the best bang for your buck out of mental ray rendering. In the previous chapter, you quickly lit the plane, candle, and decorative box with key, fill, and rim lights (the three-point lighting system) and set it to render with Maya Software raytracing. In this exercise, you'll take this concept a few steps further and change the scene to best use mental ray, so take a deep breath, call your mother and tell her you love her, and let's get started!

Set your project to the Plane project. Open the planeLightingScene_v06.ma file in the scenes folder. Remember that you turned off the plane's display layer earlier, so the plane will not be seen when you load this file. This scene has the lights and soft shadows created in Chapter 10 and is set to render through mental ray with raytracing enabled already. Let's take a critical look at the render from the earlier Figure 11.86.

Now that you have created a more detailed box, let's update the reference in this scene. Choose File - Reference Editor. You are currently reading in boxDetail01.mb. Click boxDetail01.mb to select the entry and choose Reference → Replace Reference. Navigate to the Decorative_Box project's scene folder and select boxDetail05.mb.

All your work detailing the box will now pop into the lamp lighting scene. Render a frame from the persp camera view. Figure 11.106 shows a comparison of the render from before (top) and the newly detailed box in the lamp lighting scene (bottom). The detailed box looks much nicer now, although you have to tweak the render quality settings in this scene later.

Adding an IBL

Here's where things get fun. You'll add an IBL node to the scene and add the living room HDR image to that IBL. First make sure you have mental ray selected as your renderer in the Render Settings window and then follow along with these steps:

1. In your already open planeLightingScene_v06.ma scene, turn on the planeLayer visibility. Render a frame in the renderCam view, and save it to the buffer in the Render View window. You can also start moving around with your persp camera view again, since you're no longer focusing on the box.



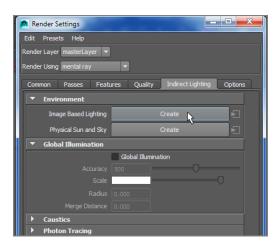
Figure 11.106

The render on the bottom shows all the added detail to the box missing from the render on the top.

2. In the Render Settings window, click the Indirect Lighting tab. At the top, in the Environment heading, click the Create button for Image Based Lighting, as shown in Figure 11.107.

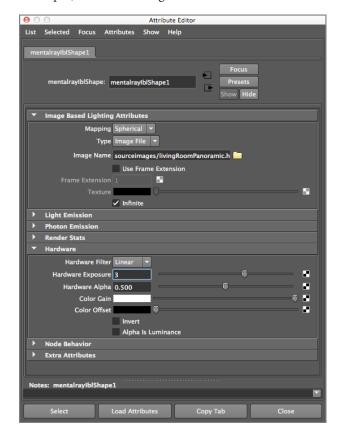
Figure 11.107
Click to create

an IBL.



3. The Attribute Editor opens to a view of the newly created IBL (named mentalray-IblShapel), as shown in Figure 11.108.

Figure 11.108
The Attribute Editor
for the IBL node



4. Click the folder icon (next to the Image Name attribute. Navigate to the sourceimages folder of the TableLamp project on your system, and select the file livingRoomPanoramic.hdr. You see in your view panes that a large yellow sphere is created in your scene. Set Hardware Exposure to 3 to lighten up the view of the HDR image in the view panels. The living-room panoramic HDR image is mapped. (See Figure 11.109.)

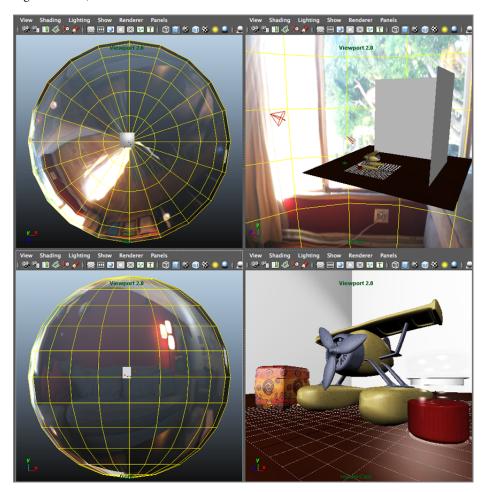


Figure 11.109 The IBL is in place

and mapped to the office chrome ball HDRI.

5. Use your persp camera to take a good look at the placement of the IBL sphere and the HDR image mapped to it. Select the IBL (you can click the IBL's yellow wireframe lines or select mentalrayIbl1 in the Outliner), and rotate the IBL sphere to 80 in the Y-axis so that the bright sun in the window is behind the renderCam and the window faces the wing of the plane on the right side of the frame. Figure 11.110 shows the orientation of the IBL.

Figure 11.110

Rotate the IBL sphere in the Y-axis to 80.



- 6. In the renderCam view, render a frame, and you'll see that the lighting is largely unchanged from your earlier renders in step 1, though you should see more reflections in the plane and the candle's glass. Next you need to enable Final Gather to take better advantage of the IBL and the HDRI mapped to it.
- 7. On the Render Settings window's Indirect Lighting tab, check the Final Gathering check box in the Final Gathering heading. Click the color swatch for Primary Diffuse Scale, and enter a V (value) of **1.5**, as shown in Figure 11.111. This will boost the brightness of the HDRI and will therefore cast more light into the scene.
- 8. Render a frame and compare it to the image in Figure 11.112. Save the image by clicking the Keep Image icon () and compare it to your earlier render before the IBL and before Final Gather. Also notice the bounced light on the redwood tabletop, which is a nice touch, albeit splotchy right now.



Figure 11.111

Enter a value of 1.5 to increase the brightness of the HDRI.

If you need to prevent the HDR image from showing up in the background of your renders, select the IBL sphere and, in the Attribute Editor under the Render Stats heading, turn off Primary Visibility. The background will then render black without the IBL.

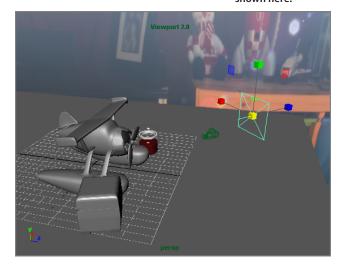
The render is a little bright and quite noisy. You have three lights in the scene already and now an HDRI. Let's get rid of all the lights and start fresh to convert this to an entire mental ray scene, shaders and all.



Figure 11.112 It's a noisy and bright render.

- 9. In the Outliner select the three lights in the scene, spotLight1, spotLight2, and directionalLight1. Press Ctrl+H to hide them, which will effectively turn them off in the scene. There's no need to delete the lights; hiding them will do.
- 10. Re-render the scene, and you'll see only the HDRI contribution to the scene. The render will get more noisy, but you'll tackle that noise a little later. You will now add some direct light with an Area light, which gives a much nicer lighting than the other lights you've used.
- 11. In the persp view panel, create an area light (Create → Lights → Area Light). Scale it to 3 in XYZ and place it to face the toy plane, as shown in Figure 11.113.
- 12. Render the scene (it will take a few minutes), and you'll see it's brighter, of course, but the toy plane looks a little bright, like the light is too close as shown in Figure 11.114. Before you change the lighting, let's re-create the materials with mental ray shaders. This will illustrate how shaders can affect lighting, and vice versa.
- 13. Open the Hypershade and MMB+drag the yellow Blinn shader to the Work Area. You'll start with changing this yellow Blinn to a yellow mia_material, a versatile mental ray shader.

Figure 11.113 Create an Area light and place it as shown here.



With the yellow Blinn selected, choose the Input and Output Connections icon () to see the connections for the Blinn shader.

Figure 11.114 Adding an Area light makes the toy airplane look a little too bright.



14. In the left side of the Hypershade, click Mental Ray Materials and click to create a mia_material_x shader, as shown in Figure 11.115.

Figure 11.115 Create an mia material_x shader.



- 15. In the Work Area of the Hypershade, MMB+drag the noiseYellow texture node from the yellow Blinn to the new mia_material_x1 shader, as shown in Figure 11.116. When you release the middle mouse button, select Diffuse from the context menu that appears. This will connect the yellow noise pattern you created earlier to the color of the new mia material. Name the new material vellowMia.
- 16. Now, let's assign the new mia material yellow to the yellow parts of the plane. This is pretty easily done through the Hypershade. RMB+click the old yellow Blinn shader and select Select Objects With Material from the marking menu. This will select all the yellow parts of the plane. Next, RMB+click the yellowMia shader and select Assign Material To Selection. That's it. See Figure 11.117.
- 17. Select the yellowMia shader and notice that the attributes are a bit different from the shaders you've seen like Blinn or Phong. The Color attribute in the Diffuse section is the basic color of the shader; that's where the yellow noise pattern is connected. You need to adjust the specular highlight to make the shader less glossy. Notice there are no specular controls. Instead, with this shader, the specular highlights are controlled in the Reflection section with the Reflectivity and Glossiness attributes. Set Glossiness to 0.35 to lessen the gloss on the yellow plastic and reduce Reflectivity to **0.20** to reduce the shine and reflection you'll get once you render.

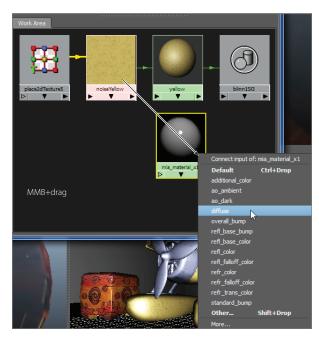


Figure 11.116 Connect the yellow texture to the new shader.

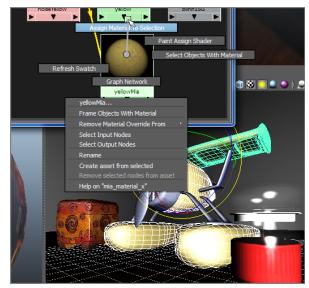


Figure 11.117

Assign the new yellowMia shader to the yellow parts of the plane.

- 18. Repeat steps 13–17 for the old gray Blinn shader to make a new gray mia_material_x (call it **greyMia**). This way you can replace the gray parts of the plane with greyMia.
- 19. Now that the toy plane has been converted to mental ray mia shaders, render the frame and compare it to your previous render with the Blinn shaders. Figure 11.118 shows the mia shader render of the plane. You can notice some subtle differences in how the mia_material is affected by light when compared to the Blinn render from Figure 11.114.



Figure 11.118

The mia material creates a more subtle look for the plane, though the scene is still too bright!

- 20. Repeat steps 13-17 again to change the shader of the redwood tabletop to a mia material as well (call it **floorMia**). Except for the floor's mia material, set Reflectivity to **0.4** and Glossiness to **0.70** to make it a bit shiny.
- 21. Lastly, make another mia material. This time, set the Diffuse section's Color to a medium gray and set Reflectivity to **0**. This will be a matte material for the walls. Assign this new shader (call it wallsMia).
- 22. Finally, select the Area light and set Intensity to 1.25 to bring down the brightness in the scene. Render and compare to previous renders and to Figure 11.119.

Lighting, rendering, and shading all go hand in hand, and as you can see from the previous exercise, you have to go back and forth to best achieve the look you want. Now you're not done yet, there is quite a bit of noise in the renders. I'll address that next. You can check your work (or skip to this point) against the scene file planeLightingScene_v07. ma from the scenes folder in the Plane project.



Figure 11.119 Rendering the mia materials on the plane, table top,

and walls

ORIENTING THE IBL

Play around with the rotations of the IBL to see how the lighting differs until you settle on a look you like.

Fixing Final Gather Noise

The render from the previous section looks nice, except that it's noisy. Notice the blotches of light all over the floor. This is from having low-quality settings for FG. On the Render Settings' Indirect Lighting tab, increase Accuracy to 750 under the Final Gathering head-

ing and Point Interpolation to 80. Doing so dramatically increases the render time for the frame (perhaps even more than doubling the previous time); however, it makes the scene look much better—practically eradicating the blotchy nature of the previous renders. (See Figure 11.120.) Save the image to your Render View buffer.





You can open the scene file planeLightingScene_v08.ma from the scenes folder in the Plane project to catch up to this point or to check your work so far.

FG is a difficult beast to tame, especially with an IBL. It takes a lot of experience, patience, and practice to be able to use this powerful feature of mental ray successfully in scenes. Experiment with different HDR images that you can download from the Internet as well as adjusting the lights in the scene.

Fixing General Render Noise

You may have gotten rid of the blotchy Final Gather noise (mostly, at least), but if you take a close look at Figure 11.120, you can see there is some noise on the back wall. This is coming from the shadow being cast by the Area light in the scene. At first blush you may think to increase the overall Render Settings Quality slider to higher than the current 0.25. And that is not a bad instinct, but this noise is all in that back shadow. If the edges of the plane or decorative box or candle were jagged, then yes, the Quality slider would be your solution. This is all about the area light and its shadow, and for that there is a focused solution.

To fix the noisy shadow, go to the planeLightingScene_v08.ma scene and select the Area light. Since this is an Area light and it inherently has soft shadows, there is no Light Radius attribute to set the shadow softness. The softness of an Area light shadow corresponds to the size of the Area light. The larger the light, the softer its shadows will render. Our shadow seems too soft, so let's make the Area light smaller first to sharpen the

Figure 11.121 Cleaner shadow!



shadow a bit.

Select the Area light and reduce its scale to 1.75 in X, Y, and Z. Next, change Shadow Rays (in the Attribute Editor, under the Shadows → Raytrace Shadow Attributes section) to 12 and render to see the shadow render much cleaner than before. Compare the render in Figure 11.121 to Figure 11.120 where Shadow Rays was set to the default of 1 and the Area light was bigger. The shadow is a bit sharper, more defined, and much cleaner.

For a nice challenge, you could re-create the shaders for the candle and decorative box using mia_materials to see the difference they can make in your renders. Many different materials could be simulated with the mia materials shader.

Adding Depth of Field

One last item of interest is adding a depth of field (DOF) to the image. This effect adds blur to the render for the areas of the image that may be out of the lens's focal depth. It can greatly add to the photo-realism of a rendered image.

Select the renderCam and open the Attribute Editor. In the Depth Of Field heading, check the Depth Of Field attribute. Set Focus Distance to 11 and set F Stop to 4. The F Stop setting, like a real lens, sets how much is in focus around the focal distance. With a higher F Stop value, the focus runs deeper than with a low F Stop value. As you adjust your Focus Distance attribute, you can see the blurriness in the camera's view panel—as long as you have Viewport 2.0 enabled, which it is by default. If not, in the view panel, choose Renderer → Viewport 2.0.

Figure 11.122 shows the final render of the scene with DOF enabled. Your render times will be longer, and if you want a better render than this one, which has some grain especially in the glass candle holder, you'll likely need to increase your Quality value. This render can take quite some time, though.

The scene planeLightingScene_v09.ma from the scenes folder in the Plane project will take you to this point.

Notice from the render that the back of the toy plane is thrown out of focus, along with the candle, decorative box, and back walls. Experiment with varying focus distance and F Stop values to try different looks for your scene.

Depending on the need and the size of the render, you'll have to set the sampling levels to suit your needs—but be warned that your rendering times will dramatically increase, easily tripling depending on the quality setsize: 480 x 270 zoom: 1.000 Render Time: 15:52 Camera: renderCam

tings. There are several ways to create DOF in a render that are faster or more controllable than what I've covered here. You can render out a depth pass to use with a lens filter in a compositing package such as Nuke to keep render times down. There are also more accurate DOF methods in mental ray and its lens shaders that you can experiment with when you've comfortably grasped the overall workflow presented here.

Try also experimenting with the different sampling modes to see whether you can shave time off the render. Finally, adding an Ambient Occlusion pass isn't absolutely necessary for this render, although it may enhance the look of the contact shadows.

Figure 11.122

The propeller is in focus! You're finished!

Wrapping Up

When you've learned a lot more about lighting and gained confidence with the processes of lighting, rendering, and compositing, you'll find a wealth of options when you render in several different layers. For example, if this plane, candle, and decorative box scene was a professional job, its render would most likely be split into a flat-color pass, a reflection pass, a specular highlight pass, a shadow pass, an Ambient Occlusion pass, a depth pass, and several different matte passes that separate parts of the scene to give the compositor the ultimate control in adjusting the image to taste.

You may not know what all those passes do right now, but understanding will come with time as long as you continue practicing your CG Kung Fu. Rendering your own color passes and Ambient Occlusions will give you some beautiful renders even without all the other passes. Keep in mind that small changes and additions can take you a long way toward a truly rich image.

For an interesting challenge, take a photo in your house; render the plane, candle, and box to match the lighting in the photo; and then composite the render against your photo.

Summary

In this chapter, you learned how to set up your scene for rendering. Starting with the Render Settings window and moving on to the different render engines available, you learned how to render your scene for a particular look. Then, I covered how to preview your render and how to use IPR for fast scene feedback. I moved on to cover how to render reflections and refractions, how to create and use cameras, and how to render with motion blur. You tested your skill on a wine bottle scene, and to batch-render it out into a sequence of images, you checked it in a program like FCheck. You also used Maya render layers, overrides, and rendering an Ambient Occlusion pass to make your renders more realistic. Finally, you applied this knowledge to rendering the toy plane, a glass candle holder, and the decorative box using Final Gather and an HDR of a living room.

Getting to this point in a scene can take some work, but when you see the results playing back on your screen, all the work seems more than worth it. Always allow enough time to ensure that your animations render properly and at their best quality. Most beginners seriously underestimate the time needed to complete this step properly in CG production.

After you create numerous scenes and render them, you'll begin to understand how to construct your next scenes so that they render better and faster. Be sure to keep on top of your file management—rendering can produce an awful lot of files, and you don't want to have them scattered all over the place.



Autodesk[®] Maya[®] Dynamics and Effects

Special effects animations simulate not only physical phenomena, such as smoke and fire, but also the natural movements of colliding bodies. Behind the latter type of animation is the Autodesk® Maya® dynamics engine, which is the sophisticated software that creates realistic-looking motion based on the principles of physics.

Another Maya animation tool, Paint Effects, can create dynamic fields of grass and flowers, a head full of hair, and other such systems in a matter of minutes. Maya also offers dynamic simulations for hair, fur, and cloth. In this chapter, I'll cover the basics of dynamics in Maya and let you practice working with particles by making steam.

Learning Outcomes: In this chapter, you will be able to

- Create rigid body dynamic objects and create forces to act upon them
- Keyframe animated passive rigid bodies to act on active rigid bodies
- Bake out simulations to keyframes and simplify animation curves
- Understand nParticle workflow
- Emit nParticles and have them move and render to simulate steam
- Draw Paint Effects strokes
- Create and manipulate an nCloth object (a.k.a. nMesh)
- Collide nMesh objects and add forces and dynamic constraints
- Cache your nDynamics to disk
- Customize the Maya interface

An Overview of Dynamics and Maya Nucleus

Dynamics is simulating motion by applying the principles of physics. Rather than assigning keyframes to objects to animate them, with Maya dynamics you assign physical characteristics that define how an object behaves in a simulated world. You create the objects as usual in Maya, and then you convert them to dynamic bodies. There are rigid bodies, which are solid objects, and soft bodies, which are malleable surfaces. Soft bodies won't be covered in this book; however, I will go through nCloth dynamics later, which is a much more powerful way of creating soft surface dynamics.

Dynamic bodies are affected by external forces called *fields*, which exert a force on them to create motion. Fields can range from wind forces to gravity and can have their own specific effects on dynamic bodies.

In Maya, dynamic objects are categorized as bodies, particles, hair, and fluids. Dynamic bodies are created from geometric surfaces and are used for physical objects such as bouncing balls. Particles are points in space that have renderable properties and are used for numerous effects, such as fire and smoke. I'll cover particle basics in the latter half of this chapter. *Hair* consists of curves that behave dynamically, such as strings. Fluids are, in essence, volumetric particles that can exhibit surface properties. You can use fluid dynamics for natural effects such as billowing clouds or plumes of smoke.

Nucleus is a more stable and interactive way of calculating dynamic simulations in Maya than its traditional dynamics engine. Nucleus speeds up the creation and increases the stability of some dynamic effects in Maya, including particle effects (nParticles) and cloth simulation (through nCloth).

I'll introduce nParticles later in this chapter; however, soft bodies, nHair, nCloth, and fluid dynamics are advanced topics and won't be covered in this book.

Rigid Bodies

Rigid bodies are solid objects, such as a pair of dice or a baseball, that move and rotate according to the dynamics applied. Fields and collisions affect the entire object and move it accordingly.

Creating Active and Passive Rigid Body Objects

Any surface geometry in Maya can be converted to a rigid body. After it's converted, that surface can respond to the effects of fields and take part in collisions. Sounds like fun, eh?

The two types of rigid bodies are active and passive. An *active rigid body* is affected by collisions and fields. A passive rigid body isn't affected by fields and remains still when it collides with another object. A passive rigid body is used as a surface against which active rigid bodies collide.

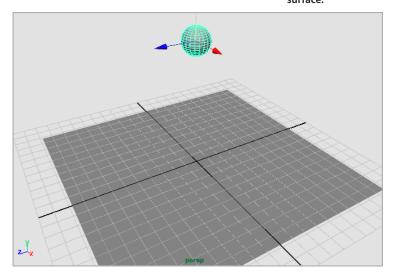
The best way to see how the two types of rigid bodies behave is to create some and animate them. In this section, you'll do that in the classic animation exercise of a bouncing ball. To create a bouncing ball using Maya rigid bodies, switch to the Dynamics menu and follow these steps:

- 1. Create a polygonal plane and scale it to be a ground surface.
- 2. Create a poly sphere and position it a number of units above the ground, as shown in Figure 12.1.
- 3. Make sure you're in the Dynamics menu set (choose Dynamics from the Status line's drop-down bar). Select the poly sphere and choose Soft/Rigid Bodies Create Active Rigid Body. The sphere's Translate and Rotate attributes turn yellow. There will be a dynamic input for those attributes, so you can't set keyframes on any of them.
- 4. Select the ground plane and choose Soft/Rigid Bodies → Create Passive Rigid Body. Doing so sets the poly plane as the floor. For this exercise, stick with the default settings and
 - ignore the various creation options and Rigid Body attributes.
- 5. To put the ball into motion, you need to create a field to affect it. Select the sphere and choose Fields Gravity. By selecting the active rigid objects while you create the field, you connect that field to the objects automatically. Fields affect only the active rigid bodies to which they're connected. If you hadn't established this connection initially, you could still do so later, through the Dynamic Relationships Editor. You'll find out more about this process later in this chapter.

If you try to scrub the timeline, you'll notice that the animation doesn't run properly. Because dynamics simulates physics, no keyframes are set. You must play the scene from start to finish for the calculations to execute properly. You must also play the scene using the Play Every Frame option. Click the Animation Options button to the right of the Range slider, or choose Window \rightarrow Settings/Preferences \rightarrow Preferences. In the Preferences window, choose Time Slider under the Settings header. Choose Play Every Frame from the Playback Speed menu. You can also set the maximum frames per second that your scene will play back by setting the Max Playback Speed attribute.

Figure 12.1

Place a poly sphere a few units above a poly plane ground surface.



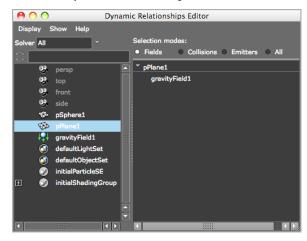
To play back the simulation, set your frame range from 1 to at least 500. Go to frame 1 and click Play. Make sure you have the proper Playback Speed settings in your Preferences window; otherwise, the simulation won't play properly.

When the simulation plays, you'll notice that the sphere begins to fall after a few frames and collides with the ground plane, bouncing back up.

As an experiment, try turning the passive body plane into an active body using the following steps:

- 1. Select the plane and open the Attribute Editor.
- 2. In the rigidBody2 tab, select the Active check box. This switches the plane from a passive body to an active body.
- 3. Play the simulation. The ball falls to hit the plane and knock it away. Because the plane is now an active body, it's moved by collisions. But because it isn't connected to the gravity field, it doesn't fall with the ball.

To connect the now-active body plane to the gravity field, open the Dynamic Relationships Editor window, shown in Figure 12.2 (choose Window → Relationship Editors - Dynamic Relationships).



On the left is an Outliner list of the objects in your scene. On the right is a list from which you can choose a category of objects to list: Fields (default), Collisions, Emitters, or All. Select the geometry (pPlanel) on the left side and then connect it to the gravity field by selecting the gravityField1 node on the right.

When you connect the gravity field to the plane and run the

simulation, you'll see the plane fall away with the ball. Because the two fall at the same rate (the rate set by the single gravity field), they don't collide. To disconnect the plane from the gravity field, deselect the gravity field in the right panel.

You can also connect a dynamic object to a field by selecting the dynamic object or objects and then the desired field and choosing Fields → Affect Selected Object(s). This method is more useful for connecting multiple dynamic objects to a field.

Turning the active body plane back to a passive floor is as simple as returning to frame 1, the beginning of the simulation, and clearing the Active attribute in the Attribute

Figure 12.2 The Dynamic **Relationships Editor** window Editor. By turning the active body back to a passive body, you regain an immovable floor upon which the ball can collide and bounce.

RELATIONSHIP EDITORS

The relationship editors, such as the Dynamic Relationships Editor window, let you connect two nodes to create a special relationship. With the Dynamic Relationships Editor window you connect dynamic attributes so that fields, particles, and rigid bodies can interact in a simulation. Another example of a relationship editor is the Light Linking window mentioned in Chapter 10, "Autodesk® Maya® Lighting," which allows you to connect lights to geometry so that they light only a specific object or objects. These are fairly advanced topics; however, as you learn more about Maya, their use will become integral in your workflow.

Moving a Rigid Body

Because the Maya dynamics engine controls the movement of any active rigid bodies, you can't set keyframes on their translation or rotation. With a passive object, however, you can set keyframes on translation and rotation as you can with any other Maya object. You also can easily keyframe an object to turn either active or passive. For instance, in the earlier bouncing ball example, you could animate the rotation of the passive body ground plane to roll the ball around on it.

Any movement that the passive body has through regular keyframe animation is translated into momentum, which is passed on to any active rigid bodies with which the passive body collides. Think of a baseball bat that strikes a baseball. The bat is a passive rigid body that you have keyframed to swing. The baseball is an active rigid body that is hit by (collides with) the bat as it swings. The momentum of the bat is transferred to the ball, and the ball is sent flying into the stadium stands. You'll see an example of this in action in the next exercise.

Rigid Body Attributes

Here is a rundown of the more important attributes for both passive and active rigid bodies as they pertain to collisions:

Mass Sets the relative mass of the rigid body. Set on active or passive rigid bodies, *mass* is a factor in how much momentum is transferred from one object to another. A more massive object pushes a less massive one with less effort and is itself less prone to movement when hit. Mass is relative, so if all rigid bodies have the same mass value, there is no difference in the simulation.

Static and Dynamic Friction Sliders Set how much friction the rigid body has while at rest (static) and while in motion (dynamic). Friction specifies how much the object resists moving or being moved. A friction of 0 makes the rigid body move freely, as if on ice.

Bounciness Specifies how resilient the body is upon collision. The higher the *bounciness* value, the more bounce the object has upon collision.

Damping Creates a drag on the object in dynamic motion so that it slows down over time. The higher the *damping*, the more the body's motion diminishes.

Initial Velocity Gives the rigid body an initial push to move it in the corresponding axis.

Initial Spin Gives the rigid body object an initial twist to start the rotation of the object in that axis.

Impulse Position Gives the object a constant push in that axis. The effect is cumulative; the object will accelerate if the impulse isn't turned off.

Spin Impulse Rotates the object constantly in the desired axis. The spin will accelerate if the impulse isn't turned off.

Center of Mass (0,0,0) Places the center of the object's mass at its pivot point, typically at its geometric center. This value offsets the center of mass, so the rigid body object behaves as if its center of balance is offset, like trick dice or a top-weighted ball.

Creating animation with rigid bodies is straightforward and can go a long way toward creating natural-looking motion for your scene. Integrating such animation into a final project can become fairly complicated, though, so it's prudent to become familiar with the workings of rigid body dynamics before relying on that sort of workflow for an animated project.

Here are a few suggestions for scenes using rigid body dynamics:

Bowling Lane The bowling ball is keyframed as a passive object until it hits the active rigid pins at the end of the lane. This scene is simple to create and manipulate.

Dice Active rigid dice are thrown into a passive rigid craps table. This exercise challenges your dynamics abilities as well as your modeling skills if you create an accurate craps table.

Game of Marbles This scene challenges your texturing and rendering abilities as well as your dynamics abilities as you animate marbles rolling into each other.

Rigid Body Dynamics: Shoot the Catapult!

Now for a fun exercise, you'll use rigid body dynamics to shoot a projectile with the already animated catapult from Chapter 8. Open the scene file catapult_anim_v2.mb from the Catapult_Anim project downloaded from the companion website.

1. For the projectile, create a polygon sphere and place it above the basket, as shown in Figure 12.3.

Since dynamics takes a lot of calculations, to make things easier you will create an invisible object to be the passive rigid body collider instead of the existing geometry of the basket itself. This is a frequent workflow in dynamics, where proxy geometry (often hidden) is used to alleviate calculations and speed up the scene.

- 2. Create another poly sphere with Axis Divisions and Height Divisions both set to **40**.
- 3. Select the top-half faces of the sphere and delete them to create a bowl and place it over the catapult basket, as shown later in Figure 12.16. Use the D key along with Point Snaps to place the bowl's pivot point on the top rim of the bowl, shown in Figure 12.4.

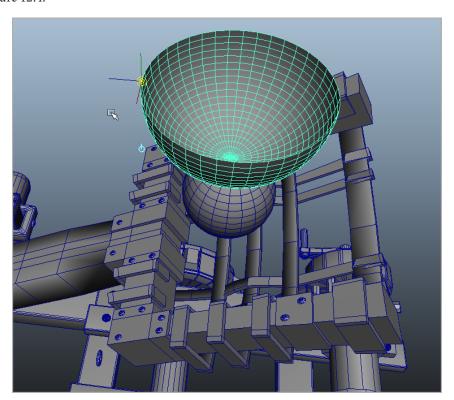


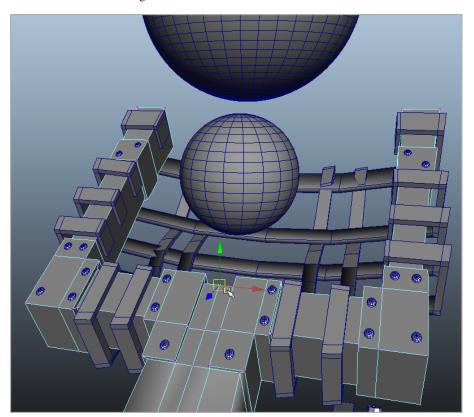
Figure 12.3 Place a sphere over the basket.

Figure 12.4 It puts the sphere in the basket, or else it gets the hose again.

The animation of the catapult arm is driven by a Bend deformer, so simply placing and grouping the bowl to the basket will not work. You will use a point on poly constraint to rivet the bowl to a vertex on the arm.

4. Select the vertex shown in Figure 12.5; then Shift+select the bowl object. In the Animation menu set, choose Constrain - Point On Poly. The bowl will snap to the arm, as shown in Figure 12.6.

Figure 12.5 Select the vertex.



- 5. Select the bowl and, in the Channel Box, click the pSphere2_pointOnPolyConstraint1 node. Set Offset Rotate Y to -90 and Offset Rotate X to 8 to place the bowl in the basket better (Figure 12.7).
- 6. Go into the Dynamics menu set. Select the projectile sphere and at frame 1 of the animation choose Fields - Gravity. The ball will instantly become an active rigid body and will have a gravity attached. If you play back your scene, the ball will simply fall through the bowl and catapult.

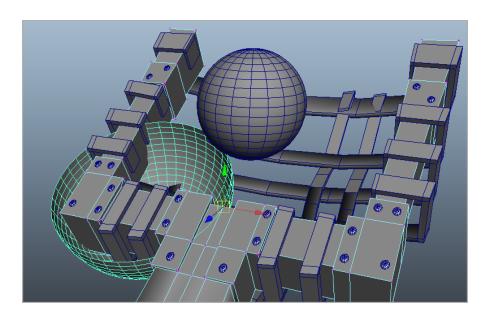


Figure 12.6 The bowl snaps to the arm at a weird

angle.

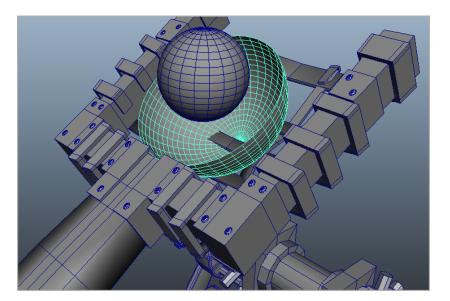


Figure 12.7 Set offsets to place the bowl properly in the basket.

7. Select the bowl object and choose Soft/Rigid Bodies - Create Passive Rigid Body to make the bowl a collision surface for the projectile ball. Play back your scene and bam! The projectile will bounce into the bowl as the catapult arm bends back, and the catapult will shoot the ball out.

8. Select the bowl object and press Ctrl+H to hide it. Now when you play back your scene, the ball looks like it rests inside the catapult basket, as shown in Figure 12.8. The scene file catapult_dynamics_v1.mb will catch you up to this point.

If you play back the scene frame by frame, you'll notice that the arm and basket will briefly pass through the projectile ball for a few frames as the arm shoots back up around frames 102–104. Increasing the subdivisions of the ball and the bowl before making them rigid bodies will help the collisions, but for a fun exercise, this works great. Play with the placement of the ball at the start of the scene, as well as the gravity and any other fields you care to experiment with, to try to land the projectile in different places in the scene.

Baking Out a Simulation

Frequently, you create a dynamic simulation to fit into another scene, perhaps to interact with other objects. In such cases, you want to exchange the

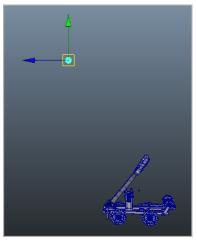


Figure 12.8 The ball shoots!

dynamic properties of the dynamic body you have set up in a simulation for regular, old-fashioned animation curves that you can more easily edit. You can easily take a simulation that you've created and bake it out to curves. As much fun as it is to think of cupcakes, baking is a somewhat catchall term used to describe converting one type of action or procedure into another; in this case, you're baking dynamics into keyframes.

You'll take the simulation you set up earlier with the catapult and turn it into keyframes. Keep in mind that you can use this introduction as a foundation for your own explorations.

To bake out the rigid body simulation of the catapult projectile, follow these steps:

- 1. Open the scene file catapult_dynamics_v1.mb from the Catapult_Anim project on the web page, or if you prefer, open your own scene from the previous exercise.
- 2. Select the projectile ball (pSphere1) and choose Edit → Keys → Bake Simulation □. In the option box, shown in Figure 12.9, set Time Range to Time Slider (which should be set to 1 to 240). This, of course, sets the range you would like to bake into curves.
- 3. Set Hierarchy to Selected and set Channels to From Channel Box. This ensures you have control over which keys are created. Make sure Keep Unbaked Keys and Disable Implicit Control are checked and that Sparse Curve Bake is turned off. Before clicking the Bake or Apply button, select the Translate and Rotate channels in the Channel Box, Click Bake,

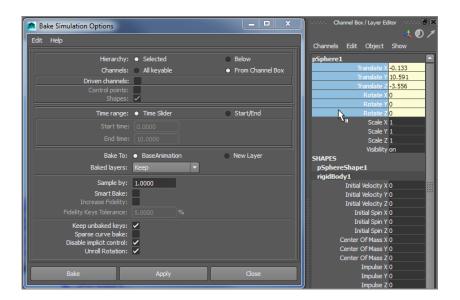


Figure 12.9 The Bake Simulation **Options window**

4. Maya runs through the simulation. Scrub the timeline back and forth. Notice how the projectile shoots as if the dynamic simulation were running—except that you can scrub in the timeline, which you can't do with a dynamics simulation. With the projectile selected, open the Graph Editor; you'll see something similar to Figure 12.10.

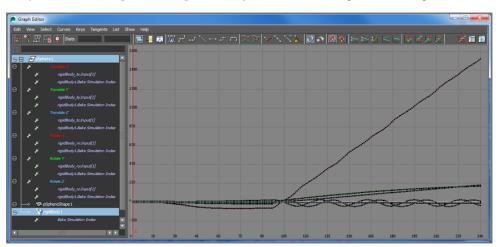


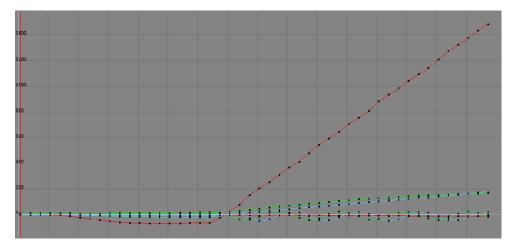
Figure 12.10 The projectile has animation curves.

The curves are crowded; they have keyframes at every frame. A typical dynamics bake gives results like this. But you can set the Bake command to sparse the curves for you; that is, it can take out keyframes at frames that have values within a certain tolerance so that a minor change in the ball's position or rotation need not have a keyframe on the curve.

5. Let's go back in time and try this again. Press Z (Undo) until you back up to right before you baked out the simulation to curves. You can also close this scene and reopen it from the original project, if necessary. This time, select the projectile sphere and choose Edit → Keys → Bake Simulation □. In the option box, turn on the Sparse Curve Bake setting and set Sample By to 5. Select the Translate and Rotate channels in the Channel Box and click Bake.

Maya runs through the simulation again and bakes everything out to curves. This time it makes a sparser animation curve for each channel because it's setting keyframes only at five-frame intervals, as shown in Figure 12.11. If you open the Graph Editor, you'll notice that the curves are much friendlier to look at and edit.

Figure 12.11 Sampling by fives makes a cleaner curve.



Sampling by fives may give you an easier curve to edit, but it may also oversimplify the animation of your objects; make sure you use the best Sampling setting for your simulation when you need to convert it to curves for editing.

Simplifying Animation Curves

Despite a higher Sampling setting when you bake out the simulation, you can still be left with a lot of keyframes to deal with, especially if you have to modify the animation extensively from here. One last trick you can use is to simplify the curve further through the Graph Editor. You have to work with curves of the same relative size, so you'll start with the rotation curves because they have larger values. To simplify the curve in the Graph Editor, follow these steps:

1. Select the projectile and open the Graph Editor. In the left Outliner side, select Rotate X, Rotate Y, and Rotate Z to display only these curves in the graph view. Figure 12.12 shows the curves.

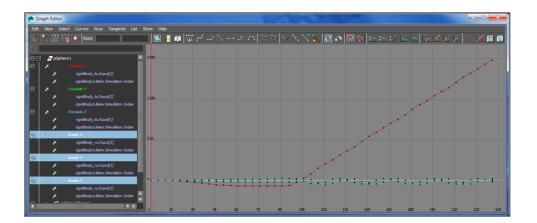


Figure 12.12

The Graph Editor displays the rotation curves of the projectile ball.

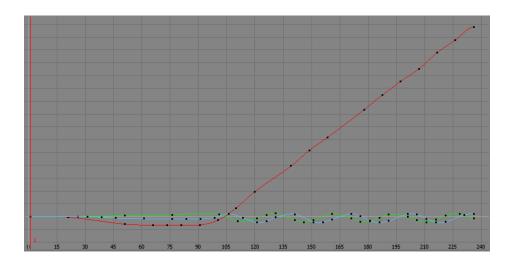
- 2. In the left panel of the Graph Editor, select the rigidBody_rx.Input[1] nodes displayed under the Rotate X, Y, and Z entries for all three curves, as shown in Figure 12.13. In the Graph Editor menu, choose Curves → Simplify Curve □. In the option box, set Time Range to All, set Simplify Method to Classic, set Time Tolerance to 10, and set Value Tolerance to 1.0. These are fairly high values, but because you're dealing with rotation of the projectile, the degree values are high enough. For more intricate values such as Translation, you would use much lower tolerances when simplifying a curve.
- rigidBody_ty.Input[1] rigidBody1.Bake Simulation Inde rigidBody1.Bake Simulation Inde

Figure 12.13 Select the curves to simplify them.

3. Click Simplify, and you see that the curves retain their basic shapes but lose some of their keyframes. Figure 12.14 shows the simplified curves, which differ perhaps a little from the original curve that had keys at every five frames.

Simplifying curves is a handy way to convert a dynamic simulation to curves. Keep in mind that you may lose fidelity to the original animation after you simplify a curve, so use this technique with care. The curve simplification works with good old-fashioned keyframed curves as well; if you inherit a scene from another animator and need to simplify the curves, do it just as you did here.

Figure 12.14 A simplified curve for the rotations of the eight ball



nParticle Dynamics

Like rigid body objects, particles are moved dynamically using collisions and fields. In short, a particle is a point in space that is given renderable properties—that is, it can render out. When particles are used en masse, they can create effects such as smoke, a swarm of insects, fireworks, and so on. nParticles are an implementation of particles through the Maya Nucleus solver, which provides better and easier simulations than traditional Maya particles.

Although particles (and nParticles) can be an advanced and involved aspect of Maya, it's important to have some exposure to them as you begin to learn Maya.

It's important to think of particle animation as manipulating a larger system rather than as controlling every single particle. Particles are most often used together in large numbers so that the entirety is rendered out to create an effect. You control fields and dynamic attributes to govern the motion of the system as a whole.

Emitting nParticles

A typical workflow for creating an nParticle effect in Maya breaks out into two parts: motion and rendering. First, you create and define the behavior of particles through emission. An emitter is a Maya object that creates the particles. After you create fields and adjust particle behavior within a dynamic simulation, much as you would do with rigid body motion, you give the particles renderable qualities to define how they look. This second aspect of the workflow defines how the particles come together to create

To create an nParticle system, follow these steps:

1. Make sure you're in the nDynamics menu set, choose nParticles → Create nParticles, and select Cloud. Choose nParticles → Create nParticles → Create Emitter □. The option box gives you various creation options for the nParticle emitter, as shown in Figure 12.15.

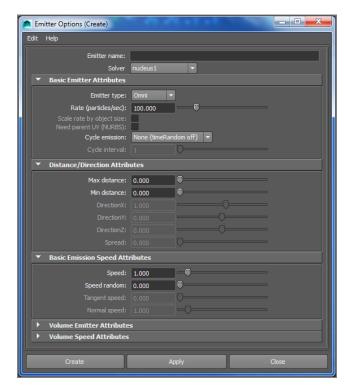


Figure 12.15
Creation options for an nParticle emitter

The default settings create an Omni emitter with a rate of 100 particles per second and a speed of 1.0. Click Create. A small round object (the emitter) appears at the origin.

2. Set your time range to 1–240 frames. Click the Play button to play the scene. As with rigid body dynamics, you must also play back the scene using the Play Every Frame option. You can't scrub or reverse-play particles unless you create a cache file. You'll learn how to create a particle disk cache later in this chapter.

You'll notice a mass of circles streaming out of the emitter in all directions (see Figure 12.16). These are the nParticles.

An Omni emitter emits a swarm of cloud particles in all directions.

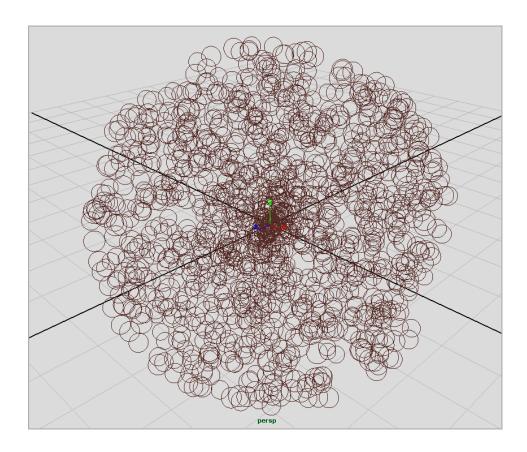
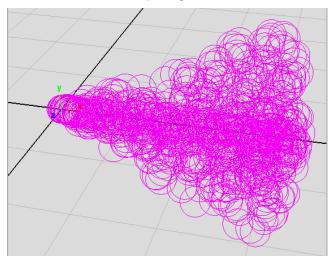


Figure 12.17
Cloud nParticles are sprayed in a specific direction.

Emitter Attributes

You can control how particles are created and behave by changing the type of emitter and adjusting its attributes. Here are the most often used emitters:



Omni Emits particles in all directions.

Directional Emits a spray of particles in a specific direction, as shown in Figure 12.17.

Volume Emits particles from within a specified volume, as shown in Figure 12.18. The volume can be a cube, a sphere, a cylinder, a cone, or a torus. By default, the particles can leave the perimeter of the volume.

Figure 12.18 **Cloud nParticles** emit from anywhere inside the emitter's volume.

After you create an emitter, its attributes govern how the particles are released into the scene. Every emitter has the following attributes to control the emission:

Rate Governs how many particles are emitted per second.

Speed Specifies how fast the particles move out from the emitter.

Speed Random Randomizes the speed of the particles as they're emitted, for a more natural look.

Min and Max Distance Emits particles within an offset distance from the emitter. You enter values for the Min and Max Distance settings. Figure 12.19 shows a Directional emitter with Min and Max Distance settings of 3.

Figure 12.19 An emitter with Min and Max Distance settings of 3 emits cloud nParticles 3 units from itself

nParticle Attributes

After being created, or born, and set into motion by an emitter, nParticles rely on their own attributes and any fields or collisions in the scene to govern their motion, just like rigid body objects.

In Figure 12.20, the Attribute Editor shows a number of tabs for the selected particle object. nParticle1 is the particle object node. This has the familiar Translate, Rotate, and Scale attributes, like most other object nodes. But the shape node, nParticleShapel, is

Figure 12.20 nParticle attributes



where all the important attributes are for a particle, and it's displayed by default when you select a particle object. The third tab in the Attribute Editor is the emitter1 node that belongs to the particle's emitter. This makes it easier to toggle back and forth to adjust emitter and particle settings.

The Lifespan Attributes

When any particle is born, you can give it a *lifespan*, which allows the particle to die when it reaches a certain point in time. As you'll see with the steam locomotive later in the chapter, a particle that has a lifespan can change over that lifespan. For example, a particle may start out as white and fade away at the end of its life. A lifespan also helps keep the total number of particles in a scene to a minimum, which helps the scene run more efficiently.

You use the Lifespan mode to select the type of lifespan for the nParticle.

Live Forever The particles in the scene can exist indefinitely.

Constant All particles die when their Lifespan value is reached. Lifespan is measured in seconds, so upon

emission, a particle with a Lifespan of 1.0 will exist for 30 frames (in a scene set up at 30fps) before it disappears.

Random Range This type sets a lifespan in Constant mode but assigns a range value via the Lifespan Random attribute to allow some particles to live longer than others for a more natural effect.

LifespanPP Only This mode is used in conjunction with expressions that are programmed into the particle with Maya Embedded Language (MEL). Expressions are an advanced Maya concept and aren't used in this book.

The Shading Attributes

The Shading attributes determine how your particles look and how they will render. Two types of particle rendering are used in Maya: software and hardware. Hardware particles are typically rendered out separately from anything else in the scene and are then composited with the rest of the scene. Because of this compound workflow for hardware particles, this book will introduce you to a software particle type called Cloud. Cloud, like other software particles, can be rendered with the rest of a scene through the software renderer.

With your particles selected, open the Attribute Editor. In the Shading section, you'll find the Particle Render Type drop-down menu (see Figure 12.21).

The three render types listed with the (s/w) suffix are software-rendered particles. All the other types can be rendered only through the Maya Hardware renderer. Select your render type, and Maya adds the proper attributes you'll need for the render type you selected.

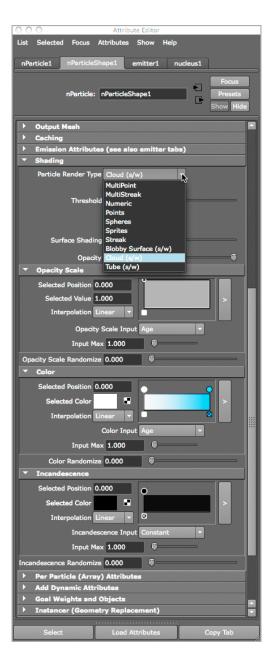
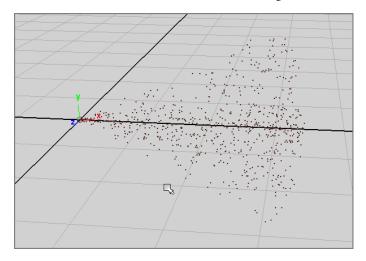


Figure 12.21

The Particle Render Type drop-down menu

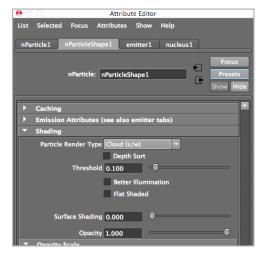
For example, if you select the Points render type from the menu, your particles change from circles on the screen to dots, as shown in Figure 12.22.

Figure 12.22 Dots on the screen represent point particles.



Several new attributes that control the look of the particles appear when you switch the Particle Render Type setting. Each Particle Render Type setting has its own set of render attributes. Set your nParticles back to the Cloud type. The Cloud particle type attributes are Threshold, Surface Shading, and Opacity. (See Figure 12.23.)

Figure 12.23 **Each Particle Render** Type setting displays its own set of attributes.



In the Shading heading, shown in Figure 12.24 for the cloud nParticles, are controls for the Opacity Scale, Color, and Incandescence attributes. They control how the particles look when simulated and rendered. Notice how each of these controls is based on ramps.

Likewise, the Particle Size heading in the Attribute Editor contains a ramp for Radius Scale that works much the same way as the Color attribute just described. In this case, you use the Radius Scale ramp to increase or decrease the size of the particle along its Age setting.

nCaching Particles

It would be nice to turn particles into money cash, but I can only show you how to turn your particles into a disk cache. You can cache the motion of your particles to memory or to disk to make playback and editing of your particle animation easier. To cache particles to your system's fast RAM memory, select the nParticle

object you want to cache, and open the Attribute Editor. In the Caching section, under Memory Caching, select the Cache Data check box. Play back your scene, and the particles cache into your memory for faster playback. You can also scrub your timeline to see your particle animation. If you make changes to your animation, the scene won't reflect the changes until you delete the cache from memory by selecting the particle object and unchecking the Memory Cache check box in the Attribute Editor. The amount of information the memory cache can hold depends on your machine's RAM.

Although memory caching is generally faster than disk caching, creating a disk cache lets you cache all the particles as they exist throughout their duration in your scene and ensures that the particles are rendered correctly, especially if you're rendering on multiple computers or across a network. You usually create a particle disk cache before rendering.

If you make changes to your particle simulation but you don't see the changes reflected when you play back the scene, make sure you've turned off any memory or deleted any disk cache from previous versions of the simulation.

Creating an nCache on Disk

After you've created a particle scene and you want to be able to scrub the timeline back and forth to see your particle motion and how it acts in the scene, you can create a particle nCache to disk. This lets you play back the entire scene as you like, without running the simulation from the start and by every frame.

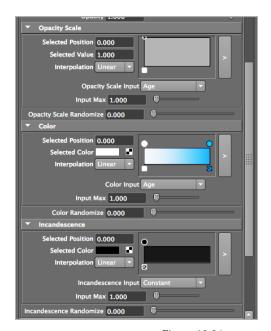


Figure 12.24
Controls for the
Opacity Scale, Color,
and Incandescence
attributes

To create an nCache, make sure to be in the nDynamics menu set, select the nParticle object in your view panel or Outliner, and choose nCache → Create New Cache. Maya will run the simulation according to the timeline and save the position of all the particle systems in the scene to cache files in your current project's Data/Cache folder. You can then play or even scrub your animation back and forth, and the particles will run properly.

If you make any dynamics changes to the particles, such as emission rate or speed, you'll need to detach the cache file from the scene for the changes to take effect. Choose nCache - Delete Cache. You can open the option box to select whether you want to delete the cache files physically or merely detach them from the current nParticles.

Now that you understand the basics of particle dynamics, it's time to see for yourself how they work.

Animating a Particle Effect: Locomotive Steam

You'll create a spray of steam puffing out of a pump on the side of a locomotive that drives the wheels that you rigged previously. You'll use the scene fancy_locomotive_anim_v3.mb from Chapter 9, "More Animation!"

Emitting the nParticles

The first step is to create an emitter to spray from the steam pump and to set up the motion and behavior of the nParticles.

- 1. In the nDynamics menu, choose nParticles Create nParticles. Cloud should still be checked in the Create nParticles menu. If it isn't, select Cloud and then choose nParticles → Create nParticles → Create Emitter □. Set Emitter Type to Directional and click Create. Place the emitter at the end of the pump, as shown in Figure 12.25.
- 2. To set up the emission in the proper direction, adjust the attributes of the emitter. In the Distance/Direction Attributes section, set Direction Y to 0, Direction X to 0, and Direction Z to 1. This emits the particles straight out of the pump over the first large wheel and toward the back of the engine.

The Direction attributes are relative. Entering a value of 1 for Direction X and a value of 2 for Direction Y makes the particles spray at twice the height (Y) of their lateral distance (X).

3. Play back your scene. The cloud nParticles emit in a straight line from the engine, as shown in Figure 12.26.

You can load the file Locomotive_Steam_v1.ma from the Locomotive project on the web page to check your work.

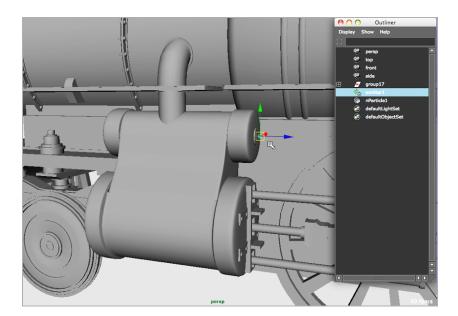


Figure 12.25

Place the emitter at the end of the pump.

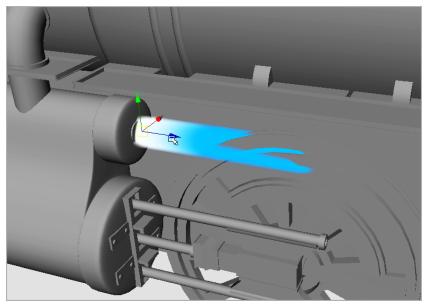


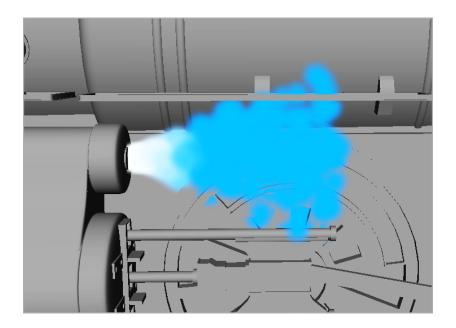
Figure 12.26

Cloud nParticles emit in a straight line from the pump.

4. To change the particle emission to more of a spray, adjust the Spread attribute for the emitter. Click the emitter1 tab in the particle's Attribute Editor (or select the emitter to focus the Attribute Editor on it instead) and change Spread from 0 to 0.30. Figure 12.27 shows the new cloud spray.

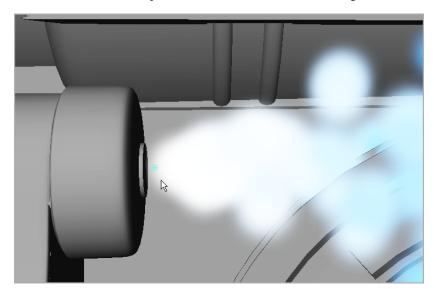
The Spread attribute sets the cone angle for a directional emission. A value of 0 results in a thin line of particles. A value of 1 emits particles in a 180-degree arc.

Figure 12.27 The emitter's Spread attribute widens the spray of particles.



- 5. The emission is rather slow for hot steam being pumped out as the locomotive drives the wheels, so change the Speed setting for the emitter from 1 to 2.0 and change Speed Random from 0 to 1. Doing so creates a random speed range between 1 and 3 for each particle. These two attributes are found in the emitter's Attribute Editor in the Basic Emission Speed Attributes section.
- 6. So that all the steam doesn't emit from the same point, keep the emitter's Min Distance at 0, but set its Max Distance to 0.3. This creates a range of offset between 0 and 0.3 units for the particles to emit from, as shown in Figure 12.28.

Figure 12.28 A range of offset between 0 and 0.3 units creates a more believable emission.



Setting nParticle Attributes

It's always good to get the particles moving as closely to what you need as possible before you tend to their look. Now that you have the particles emitting properly from the steam pump, you'll adjust the nParticle attributes. Start by setting a lifespan for them and then add rendering attributes.

- 1. Select the nParticle object and open the Attribute Editor. In the Lifespan section, set Lifespan Mode to Random Range. Set Lifespan to 2 and Lifespan Random to 1. This creates a range of 1 to 3 for each particle's lifespan. (This is based on a lifespan of 2, plus or minus a random value from 0 to 1.)
- 2. You can control the radius of the particles as they're emitted from the pump. Under the Particle Size heading in the Attribute Editor, set a Radius attribute of **0.50**. Set the Radius Scale Randomize attribute to 0.25. This allows you to have particles that emit with a radius range between 0.25 and 0.75. Figure 12.29 shows the attributes.
- 3. If you play back the simulation, you'll see the particles are quite large initially. Because steam expands as it travels, you need to adjust the size (radius) of your nParticles to make them smaller at birth; they will grow larger over their lifespan and produce a good look for your steam.
- 4. In the Radius Scale heading in the Attribute Editor, click the arrow to the right of the ramp to open a larger view of the ramp, as shown in Figure 12.30.
- 5. Click and drag the ramp's first and only handle (an open circle at the upper-left corner of the Ramp window) down to a value of about 0.25, as shown in Figure 12.31. The particles get smaller as you adjust the ramp value.

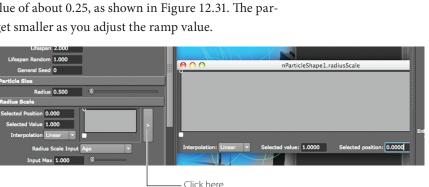


Figure 12.29 The initial radius settings for the



Figure 12.30 Open the Radius Scale ramp.

Figure 12.31

Decrease the radius of the nParticles using the Radius Scale ramp.

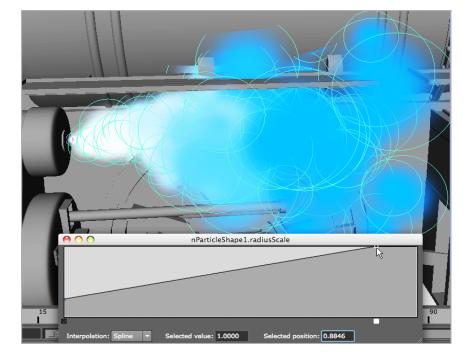


6. To allow the particles to grow in size, add a second handle to the scale curve by clicking anywhere on that line and drag to a value of 1.0 and the position shown in Figure 12.32. The particles toward the end of the spray get larger.

Figure 12.32

Adjust the Radius

Scale ramp to grow
the particles.



7. You can set up collisions so the nParticle steam doesn't travel right through the mesh of the locomotive. Select the meshes shown in Figure 12.33 and choose nMesh -Create Passive Collider.

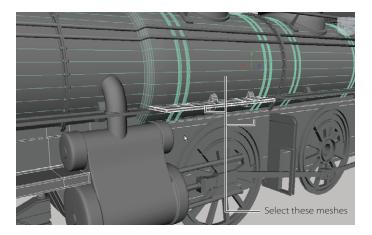


Figure 12.33 Select the meshes with which the nParticles will

collide.

8. In the Outliner, select the three nRigid nodes that were just created and set the Friction attribute in the Channel Box to 0.0 from the default of 0.10. Play back your scene, and your particles now collide with the surface of the locomotive. (See Figure 12.34.)

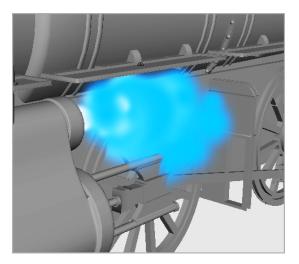


Figure 12.34 The particles now react nicely against the side of the locomotive.

If you want to check your work, download the file locomotive_steam_v2.ma from the Locomotive project on the web page.

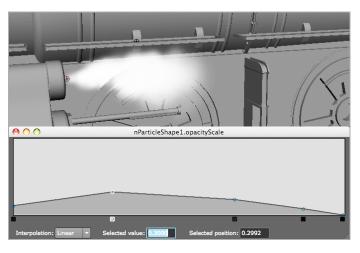
Setting Rendering Attributes

After you define the nParticle movement to your liking, you can create the proper look for the nParticles. This means setting and adjusting their rendering parameters.

PARTICLE TYPE

Because Maya has several types of particles, the particles are set up according to their type; the workflow in this section applies only to the Cloud particle type.

- 1. Select the nParticle object and open the Attribute Editor. Expand the Shading heading, and look at the Color ramp. By default, the particles go from white to light blue in color. Click the cyan color's circle handle on top of the ramp. The Selected Color attribute next to the ramp shows the cyan color. Click in the swatch to open the Color Chooser and change the color to a very light gray.
- 2. Click the arrow bar to the right of the Opacity Scale ramp to open a larger ramp view. Grab the first handle in the upper-left corner and drag it down to a value of **0.12**.
- 3. The steam needs to be less opaque at its birth, grow more opaque toward the middle of its life, and fade completely away at the end of the particle's life. Click to create new handles to create a curve for the ramp, as shown in Figure 12.35. The values for the five ramp handles shown from left to right are 0.12, 0.30, 0.20, 0.08, and 0.



4. In the Render Settings window, set Image Size to 640×480. On the Maya Software tab, set Quality to Intermediate Quality. Run the animation, and stop it when some steam has been emitted. Render a frame. It should look like Figure 12.36. The steam

doesn't travel far enough along the engine; it disappears too soon.

Figure 12.35 **Creating an Opacity** ramp for the steam

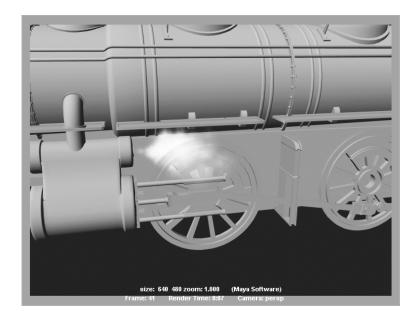


Figure 12.36 The steam seems too short and small.

With the steam nParticles selected, open the Attribute Editor; in the Radius Scale, under the Particle Size heading, change the first handle's value from 0.25 to 0.35 to make the steam particles a bit larger. Below the Opacity Scale ramp is the Input Max slider; set that value to 1.6.

5. Select the emitter1 tab, and change the Spread value for the emitter to **0.2**. Change Rate (Particles/Sec) to 200 and Speed to 3.0. Play back the simulation, and render a frame to compare to Figure 12.37.

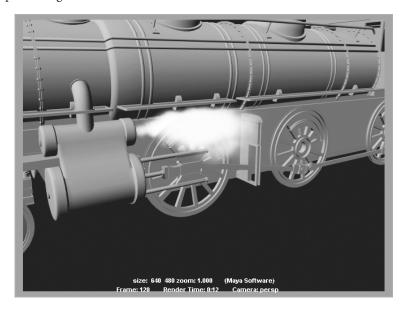
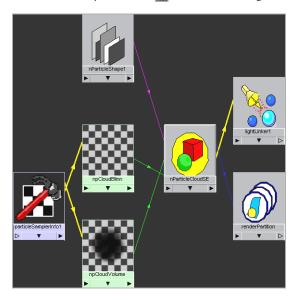


Figure 12.37

A better emission, but a bit too solid

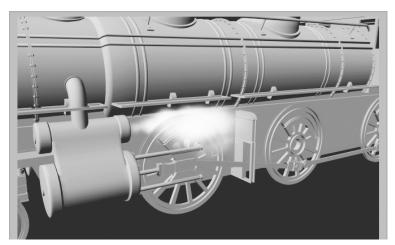
6. The Color ramp in the nParticle's Attribute Editor controls the color of the steam, and a shader is assigned to the particles. Select the steam nParticle object and open the Hypershade window. In the Hypershade, click the Graph materials on the selected object's icon (), as shown in Figure 12.38.

Figure 12.38 The shaders assigned to the cloud nParticle



- 7. In the Hypershade, select the npCloudVolume shader and open the Attribute Editor. Notice that the attributes in the Common Material Attributes section all have connections. The ramp controls in the nParticle Attribute Editor are controlling the attributes in this Particle Cloud shader.
- 8. Under the Transparency heading in the Attribute Editor for the shader, set Density to **0.35**. Try a render; the steam should look better. (See Figure 12.39.)

Figure 12.39 The steam is less flat and solid.



Batch-render a 200-frame sequence of the scene at a lower resolution, such as 320×240, to see how the particles look as they animate. (Check the frames with FCheck. Refer to Chapter 11, "Autodesk" Maya® Rendering," for more on FCheck.)

Open the file locomotive_steam_v3.ma from the Locomotive project on the web page to check your work.

Experiment with the steam by animating the Rate attribute of the emitter to make the steam pump out in time with the wheel arm. Also, try animating the Speed values and playing with different values in the Radius and Opacity ramps. The steam you'll get in this tutorial looks pretty good, but it isn't as lifelike as it could be. Particle animators are always learning new tricks and expanding their skills, and that comes from always trying new things and retrying the same effects with different methods.

When you feel comfortable with the steam exercise, try using the cloud nParticle to create steam for a mug of coffee. That steam moves much more slowly and is less defined than the blowing steam of the locomotive, and it should pose a new challenge. Also try your hand at creating a smoke trail for a rocket ship, a wafting stream of cigarette smoke, or even the billowing smoke coming from the engine's chimney.

Cloud nParticles are the perfect particle type with which to begin. As you feel more comfortable animating with clouds, experiment with the other render types. The more you experiment with all the types of nParticles, the easier they will be to harness.

Introduction to Paint Effects

One of the tools in the effects arsenal of Maya is called Paint Effects. Using Paint Effects, you can create a field of grass rippling in the wind, a head of hair or feathers, or even a colorful aurora in the sky. Paint Effects is a rendering effect found in the Rendering menu. It has incredible dynamic properties that can make leaves rustle or trees sway in a storm. Paint Effects uses its own dynamics calculations to create natural motion. It's one of the most powerful tools in Maya, with features that go far beyond the scope of this introductory book. Here you'll learn how to create a Paint Effects scene and how to access all the preset brushes to create your own effects.

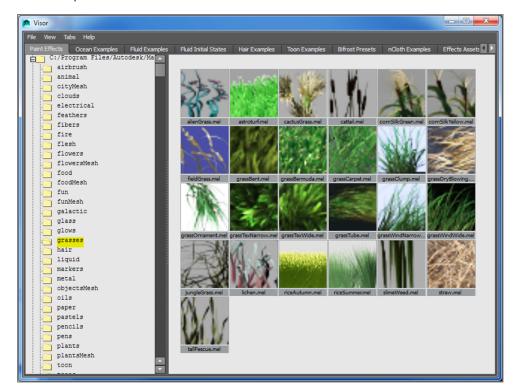
Paint Effects uses brushes to paint effects into your 3D scene. The brushes create strokes on a surface or in the Maya modeling views that produce tubes, which render out through the Maya Software renderer. These Paint Effects tubes have dynamic properties, which means they can move according to their own forces. Therefore, you can easily create a field of blowing grass.

Try This Create a field of blowing grass and flowers; it will take you all of five minutes.

1. Start with a new scene file. Maximize the perspective view. (Press the spacebar with the Perspective window active.)

2. Switch to the Rendering menu set (press F6). Choose Paint Effects → Get Brush to open the Visor window. The Visor window displays all the preset Paint Effects brushes that automatically create certain effects. Select the Grasses folder in the Visor's left panel to display the grass brushes available (see Figure 12.40). You can navigate the Visor window as you would navigate any other Maya window, using the Alt key and the mouse buttons.

Figure 12.40 The preset grass brushes in the Visor window



- 3. Click the grassWindWide.mel brush to activate the Paint Effects tool and set it to this grass brush. Your cursor changes to a Pencil icon.
- 4. In the Perspective window, click and drag two lines across the grid, as shown in Figure 12.41, to create two Paint Effects strokes of blowing grass. If you can't see the grass in your view panel, increase Global Scale in the Paint Effects Brush Settings in either the Attribute Editor or the Channel Box to see the grass being drawn onto the screen.
- 5. To change your brush so you can add some flowers between the grass, choose Paint Effects - Get Brush and select the Flowers folder. Select the dandelion_Yellow.mel brush. Your Paint Effects tool is now set to paint yellow flowers.

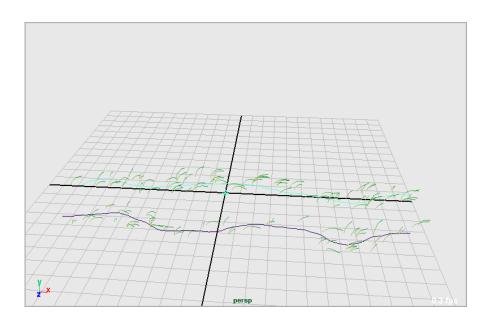


Figure 12.41 Click and drag two lines across the grid.

6. Click and drag a new stroke between the strokes of grass, as shown in Figure 12.42.

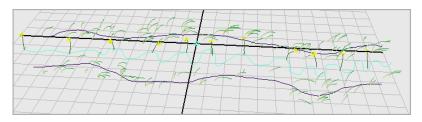


Figure 12.42

Add new strokes to add flowers to the grass.

7. Position your camera, and render a frame. Make sure you're using Maya Software and not mental ray* to render through the Render Settings window and that you use a large enough resolution, such as 640×480, so that you can see the details. Render out a 120-frame sequence to see how the grass animates in the wind. Figure 12.43 shows a scene filled with grass strokes as well as a number of different flowers.

After you create a Paint Effects stroke, you can edit the look and movement of the effect through the Attribute Editor. You'll notice, however, that there are a large number of attributes to edit with Paint Effects. The next section introduces the attributes that are most useful to the beginning Maya user.

Figure 12.43 Paint Effects can add flowers and grass to any scene.



Paint Effects Attributes

It's best to create a single stroke of Paint Effects in a blank scene and experiment with adjusting the various attributes to see how they affect the strokes. Select the stroke and open the Attribute Editor. Switch to the stroke's tab to access the attributes. For example, for an African Lily Paint Effects stroke, the Attribute Editor tab is called africanLily1.

Each Paint Effects stroke produces tubes that render to create the desired effect. Each tube (you can think of a tube as a stalk) can grow to have branches, twigs, leaves, flowers, and buds. Each section of a tube has its own controls to give you the greatest flexibility in creating your effect. As you experiment with Paint Effects, you'll begin to understand how each attribute contributes to the final look of the effect.

Here is a summary of some Paint Effects attributes:

Brush Profile Gives you control over how the tubes are generated from the stroke; this is done with the Brush Width attribute. This attribute makes tubes emit from a wider breadth from the stroke to cover more of an area.

Shading and Tube Shading Gives you access to the color controls for the tubes on a stroke.

Color 1 and Color 2 From bottom to top, graduates from Color 1 to Color 2 along the stalk only. The leaves and branches have their own color attributes, which you can display by choosing Tubes → Growth.

Incandescence 1 and 2 Adds a gradient self-illumination to the tubes.

Transparency 1 and 2 Adds a gradient transparency to each tube.

Hue/Sat/Value Rand Adds some randomness to the color of the tubes.

In the Tubes section, you'll find all the attributes to control the growth of the Paint Effects effect. In the Creation subsection, you can access the following:

Tubes Per Step Controls the number of tubes along the stroke. For example, this setting increases or decreases the number of flowers for the africanLily1 stroke.

Length Min/Max Controls the height of the tubes to make taller flowers or grass (or other effects).

Tube Width 1 and Width2 Controls the width of the tubes (the stalks of the flowers).

In the Growth subsection, you can access controls for branches, twigs, leaves, flowers, and buds for the Paint Effects strokes. Each attribute in these sections controls the number, size, and shape of those elements. Although not all strokes in Paint Effects create flowers, all strokes contain these headings.

The Behavior subsection contains the controls for the dynamic forces affecting the tubes in a Paint Effects stroke. Adjust these attributes if you want your flowers to blow more in the wind.

Paint Effects are rendered as a postprocess, which means they won't render in reflections or refractions as is and they will not render in mental ray without conversion to polygons. They're processed and rendered after every other object in the scene is rendered out in Maya Software rendering only.

To render Paint Effects in mental ray, you can convert Paint Effects to polygonal surfaces. They will then render in the scene along with any other objects so that they may take part in reflections and refractions. To convert a Paint Effects stroke to polygons, select the stroke and choose Modify → Convert → Paint Effects To Polygons. The polygon Paint Effects tubes can still be edited by most of the Paint Effects attributes mentioned so far; however, some, such as color, don't affect the poly tubes. Instead, the color information is converted into a shader that is assigned to the polygons. It's best to finalize your Paint Effects strokes before converting to polygons to avoid any confusion.

Paint Effects is a strong Maya tool, and you can use it to create complex effects such as a field of blowing flowers. A large number of controls to create a variety of effects come with that complexity. Fortunately, Maya comes with a generous sampling of preset brushes. Experiment with a few brushes and their attributes to see what kinds of effects and strange plants you can create.

Getting Started with nCloth

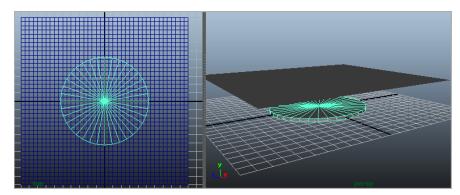
nCloth, part of the nucleus dynamics in Autodesk Maya, is a simple yet powerful way to create cloth simulations in your scene. From complex clothing on a moving character to a simple flag, nCloth dynamics can create stunning movement, albeit with some serious setup. I will very briefly touch on the nCloth workflow here to give you a taste for it and familiarize you with the basics of getting started.

Making a Tablecloth

Switch to the nDynamics menu set. The nMesh menu is where you need to start to drape a simple tablecloth on a round table in the following steps:

- 1. Create a flat disc with a poly cylinder to make the basic table. Set the Y scale to .175 and the X and Z scales to **5.25**. Set Subdivisions Axis to **36** for more segments to make the cylinder smoother.
- 2. Create a poly plane with a scale of **20** to be larger than the cylinder. Set both the Subdivisions Width and Height to 40 for extra detail in the mesh. Place the square a few units above the cylinder, as shown in Figure 12.44.

Figure 12.44 Place a square plane above a flat cylinder.



- 3. Now you'll set the plane to be a cloth object. Select the plane and choose nMesh → Create nCloth. The plane will turn purple, and a pair of new nodes is added to the scene: nucleus1 and nCloth1.
- 4. Set your frame range to **1-240**, and click Play. The plane falls down and through the cylinder object. The nucleus engine has already set up the nCloth plane to have gravity.
- 5. Now let's define the cylinder as a collision object to make it the tabletop for our tablecloth. Select the cylinder and choose nMesh → Create Passive Collider. That's it! Click Play, and watch the cloth fall onto the table and take shape around it. Figure 12.45.

Well, as easy as that was, there's a lot more to it to achieve the specific effect you may need. But you're on the road already. The first thing to understand is that the higher the resolution of the poly mesh, the better the look and movement of the cloth, at the cost of speed. At 40 subdivisions on the plane, the simulation runs pretty well, but you can see jagged areas of the tablecloth, so you would need to start with a much higher mesh for a smoother result.

Select the tablecloth and open the Attribute Editor. A wide range of attributes can be adjusted for different cloth settings; however, you can choose from a number of built-in presets to make life easier. In the Attribute Editor, choose Presets*, as shown in

Figure 12.46. Select Silk - Replace. As you can see, several Blend options appear in the submenu, allowing you to blend your current settings with the preset settings.

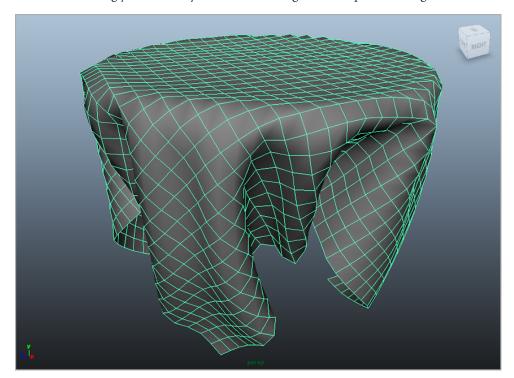


Figure 12.45 The cloth is working already!

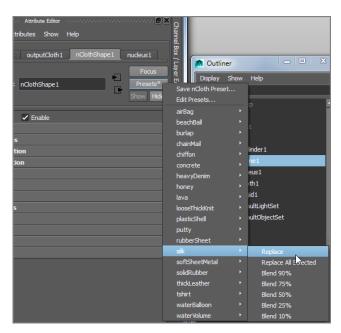
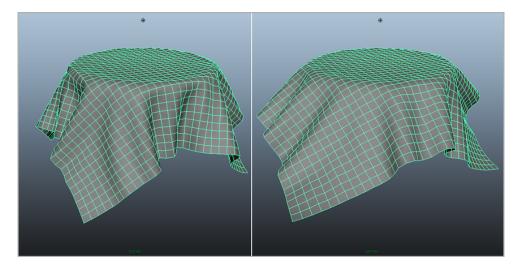


Figure 12.46 The built-in nCloth presets

In this case, you've chosen Replace to set the tablecloth object to simulate silk. It will be lighter and airier than before. Figure 12.47 (left) shows the tablecloth at frame 200 with the silk preset. Now, in the Attribute Editor, choose Preset → thickLeather → Replace for a heavier cloth simulation (Figure 12.47, right). Notice the playback for the heavy leather was slower as well.

Figure 12.47 The silk cloth simulation is lighter (left); leather simulation is heavier and runs slower.



As a beginner to Maya, you may find it most helpful to go through the range of presets and see how the attributes for the nCloth change. Here is a preliminary rundown of some of the nCloth attributes found under the Dynamic Properties heading in the Attribute Editor. Try changing some of these values to see how your tablecloth reacts, and you'll gain a finer appreciation for the mechanics of nCloth.

Stretch Resistance Controls how easily the nCloth will stretch. The lower this value, the more rubbery and elastic the cloth will behave.

Bend Resistance Controls how well the mesh will bend in reaction to external dynamic forces and collisions. The higher this value, the stiffer the cloth.

Rigidity Sets the stiffness of the cloth object. Even a small value will have a stiffening impact on the simulation.

Input Mesh Attract Compels the cloth mesh to reassume its original shape before the cloth simulation, which in our case is a flat square plane. At a value of 1, the cloth may not move at all. Using some Input Mesh Attract will also give the cloth object a stiffer and more resilient/rubbery appearance.

Figure 12.48

the flag.

Create the mesh for

Making a Flag

Now let's make a quick flag simulation to get familiar with nConstraints.

- 1. Make a rectangular polygon plane that is scaled to (20, 14, 14) and with subdivisions of 40 in width and height. Orient and place it above the home grid, as shown in Figure 12.48.
- 2. With the flag selected, choose nMesh → Create nCloth. Set your frame range to 1-240, and click Play. The flag falls straight down. You need to keep one end from falling to simulate the flag being attached to a pole.
- 3. Select the edge vertices on the left side of the flag, as shown in Figure 12.49, and choose nConstraint - Transform. The vertices you selected turn green. Now click Play, and the flag falls, while one end is tethered as if on a pole (Figure 12.50).

Select these vertices

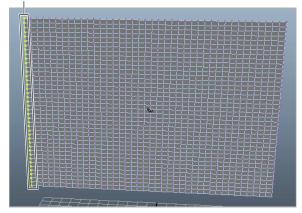


Figure 12.49 Select the vertices for the nConstraint.

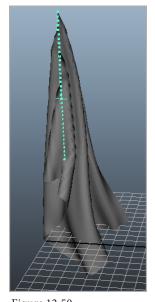


Figure 12.50 The flag is tethered to an imaginary pole now.

4. Now you'll add wind to make the flag wave. Select the flag and open the Attribute Editor. Click the nucleus1 tab, and under the Gravity and Wind heading, set the Wind Speed attribute to **16**. Figure 12.51 shows the flag flapping in the wind.

Adjust the Air Density, Wind Speed, Wind Direction, and Wind Noise attributes to adjust how the flag waves. You can use a similar workflow to create drapes blowing in an open window, for instance.

Caching an nCloth

Making a disk cache for a cloth simulation is important for better playback in your scene. Also, if you are rendering, it's always a good idea to cache your simulation to avoid any issues. Caching an nCloth is very simple. Select the cloth object, such as the flag from the previous example, and in the main menu bar, choose nCache → Create New Cache □.

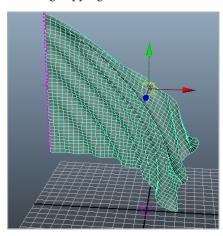
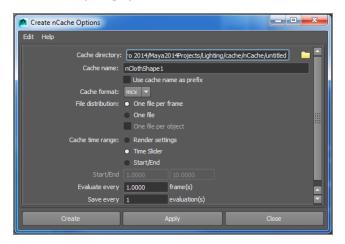


Figure 12.51 Wave the flag.

Figure 12.52 shows the options for the nCache. Here you can specify where the cache files are saved as well as the frame range for the cache. Once you create the cache, you will be able to scrub your playback back and forth.

Figure 12.52 Create nCache Options dialog box



If you need to change your simulation, you must first remove the cache before changing nCloth or nucleus attributes. To do so, select the cloth object and choose nCache → Delete Cache □. In the option box, you can select whether you want to delete the cache files or just disconnect them from the nCloth object.

You can reattach an existing cache file by selecting the nCloth object and choosing nCache – Attach Existing Cache File. The existing cache must be from that object or one with the same topology.

Customizing Maya

One of the most endearing features of Maya is its almost infinite customizability. Everyone has different tastes, and everyone works in their own way. Simply put, for everything you can do in Maya, you have several ways of doing it. There are always a couple ways to access the Maya tools, features, and functions as well.

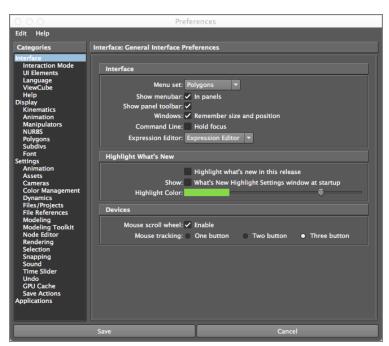
This flexibility may be confusing at first, but you'll discover that in the long run it's very advantageous. The ability to customize enables the greatest flexibility in individual workflow.

It's best to use Maya at its defaults as you first learn. However, when you feel comfortable enough with your progress, you can use this section to change some of the interface elements in Maya to better suit how you like to work.

User Preferences

All the customization features are found under Window → Settings/Preferences, which displays the window shown in Figure 12.53.

The Preferences window (see Figure 12.54) lets you make changes to the look of the program as well as to toolset defaults by selecting from the categories listed in the left pane of the window.



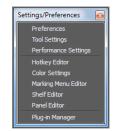


Figure 12.53
The Settings/
Preferences menu

Figure 12.54
The Preferences

window

The Preferences window is separated into categories that define different aspects of the program. Interface and Display deal with options to change the look of the program. Interface affects the main user interface, whereas Display affects how objects are displayed in the workspace.

The Settings category lets you change the default values of several tools and their general operation. An essential aspect of this category is Working Units; these options set the working parameters of your scene (in particular, the Time setting).

By adjusting the Time setting, you tell Maya your frame rate of animation. If you're working in film, you use a frame rate of 24 frames per second (fps). If you're working in NTSC video (the standard video/television format in the Americas), you use the frame rate of 30fps.

The Applications category lets you specify which applications you want Maya to start automatically when a function is called. For example, while looking at the Attribute Editor for a texture image, you can click a single button to open that image in your favorite image editor, which you specify here.

In the Settings section, click the Undo header and set Undo Queue to Infinite. Doing so allows you to undo as many actions as have occurred since you loaded the file. This feature is unbelievably handy, especially when you're first learning Maya.

Figure 12.55 The Shelf Editor



Shelves

Under the Shelf Editor command (Window → Settings/Preferences → Shelf Editor) lurks a window that manages your shelves (see Figure 12.55). You can create or delete shelves or manage the items on the Shelf with this function. This is handy when you create your own workflow for a project. Simply click the Shelves tab to display the icons on that Shelf in the Shelf Editor window. Click in the Shelf Contents section to edit the icons and where they reside on that selected Shelf. Clicking the Command tab gives you access to the MEL command for that icon when it is single-clicked in the

Shelf. Click the Double Click Command tab for the MEL command for the icon when it is double-clicked in the Shelf.

You can also edit Shelf icons from within the UI without the Shelf Editor window. To add a menu command to the current Shelf, hold down Ctrl+Alt+Shift and click the function or command directly from its menu. Items from the Tool Box, pull-down menus, or the Script Editor can be added to any Shelf.

- To add an item from the Tool Box, MMB+drag its icon from the Tool Box into the appropriate Shelf.
- To add an item from a menu to the current Shelf, hold down the Ctrl+Alt+Shift keys while selecting the item from the menu.
- To add an item (a MEL command) from the Script Editor, highlight the text of the MEL command in the Script Editor and MMB+drag it onto the Shelf. A MEL icon will be created that will run the command when you click it.
- To remove an item from a Shelf, MMB+drag its icon to the Garbage Can icon at the end of the Shelf or use Window ¬ Settings/Preferences ¬ Shelf Editor.

Hotkeys

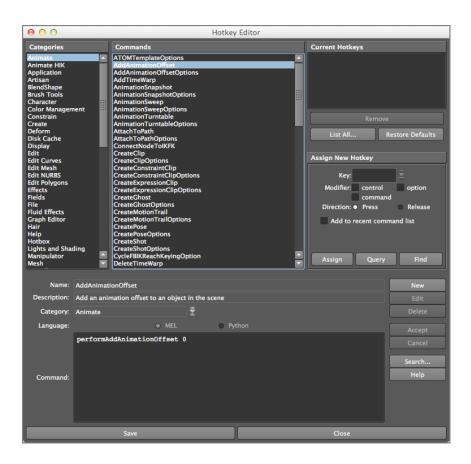
Hotkeys are keyboard shortcuts that can access almost any Maya tool or command. You've already encountered a few in your exploration of the interface and in the solar system exercise in Chapter 2, "Jumping into Basic Animation Headfirst." What fun! You can create even more hotkeys, as well as reassign existing hotkeys, through the Hotkey Editor, shown in Figure 12.56 (Window – Settings/Preferences – Hotkey Editor).

Through this monolith of a window, you can set a key combination to be used as a shortcut to virtually any command in Maya. This is the last customization you want to touch. Because so many tools have hotkeys assigned by default, it's important to get to know them first before you start changing things to suit how you work.

Every menu command is represented by menu categories on the left; the right side allows you to view the current hotkey or assign a new hotkey to the selected command. Ctrl and Alt key combinations may be used with any letter keys on the keyboard. Keep in mind that Maya is *case sensitive*, meaning that it differentiates between uppercase and lowercase letters. For example, one of my personal hotkeys is Ctrl+H to hide the selected object from view; Shift+Ctrl+H unhides it. (I'm sharing.)

The lower section of this window displays the MEL command that the menu command invokes. It also allows you to type in your own MEL commands, name them as new commands, categorize them with the other commands, and assign hotkeys to them.

Figure 12.56
The Hotkey Editor



The important thing to focus on right now is discovering how to use the tools to accomplish the tasks you need to perform and establishing a basic workflow. Toward that end, I strongly suggest learning Maya in its default configuration and only using the menu structure and default shelves to access all your commands at first, with the exception of the most basic hotkeys.

Color Settings

You can set the colors for almost any part of the interface to your liking through the Colors window shown in Figure 12.57 (Window → Settings/Preferences → Color Settings).

The window is separated into different aspects of the Maya interface by headings. The 3D Views heading lets you change the color of all the panels' backgrounds. For example, color settings give you a chance to set the interface to complement your office's decor as well as make certain items easier to read.

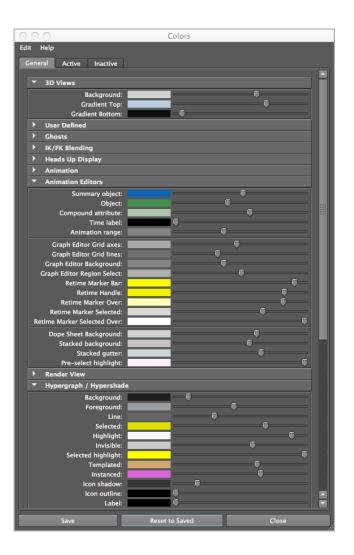


Figure 12.57
Changing the interface colors is simple.

Customizing Maya is important. However—and this can't be stressed enough—it's important to get your bearings with default Maya settings before you venture out and change hotkeys and such. When you're ready, this section of this chapter will still be here for your reference.

Summary

In this chapter, you learned how to create dynamic objects and create simulations. Beginning with rigid body dynamics, you had a quick, fun exercise where you shot a projectile with the catapult, and then you learned how to bake that simulation into

animation curves for fine-tuning. Next you learned about particle effects by creating a steam effect for a locomotive using nParticles. Next, you learned a little about the Maya Paint Effects tool and how you can easily use it to create various effects such as grass and flowers. Then you learned how to create cloth effects using nCloth to make a tablecloth and a flag, and finally, you learned how to customize Maya to suit your own preferences.

To further your learning, try creating a scene on a grassy hillside with train tracks running through. Animate the locomotive, steam and all, driving through the scene and blowing the grass as it passes. You can also create a train whistle and a steam effect when the whistle blows, and you can create various other trails of smoke and steam as the locomotive drives through.

The best way to be exposed to Maya dynamics is simply to experiment once you're familiar with the general workflow in Maya. You'll find that the workflow in dynamics is more iterative than other Maya workflows because you're required to experiment frequently with different values to see how they affect the final simulation. With time, you'll develop a strong intuition, and you'll accomplish more complex simulations faster and with greater effect.

Where Do You Go from Here?

It's so hard to say goodbye! But this is really a "hello" to learning more about animation and 3D!

Please explore other resources and tutorials to expand your working knowledge of Maya. Several websites contain numerous tips, tricks, and tutorials for all aspects of Maya; my own resources and instructional videos and links are online at http://koosh3d.com/ and on Facebook at facebook.com/IntroMaya, and you can contact or follow me through Twitter at @Koosh3d.

Of course, www.autodesk.com/maya has a wide range of learning tools. Now that you've gained your all-important first exposure, you'll be better equipped to forge ahead confidently.

The most important thing you should have learned from this book is that proficiency and competence with Maya come with practice, but even more so from your own artistic exploration. Treat this text and your experience with its information as a formal introduction to a new language and way of working for yourself; doing so is imperative. The rest of it—the gorgeous still frames and eloquent animations—come with furthering your study of your own art, working diligently to achieve your vision, and having fun along the way. Enjoy, and good luck.

Index

Note to the Reader: Throughout this index **boldfaced** page numbers indicate primary discussions of a topic. *Italicized* page numbers indicate illustrations.

	A curves, 528–529 , <i>529–530</i>
Accuracy setting, 481 , 481	flying text, 320–322 , <i>320–323</i>
Active attribute, 520	through lattices, 170–172 , <i>171–173</i>
active rigid bodies, 518–521 , <i>519–520</i>	lights, 431 , <i>431</i>
Add Attribute window, 372, 373	particle effects, 538–547 , 539–546
Add Divisions tool	relationships in
decorative box, 502–504, 503	constraints, 366–371 , <i>367–370</i>
hand, 101 , <i>101</i>	set driven keys, 371–374 , <i>371–374</i>
Add New Toon Outline option, 292	replacing objects in, 318–319 , <i>319</i>
aim constraints, 368–369 , <i>369</i>	rigging. See rigging
Aim Vector setting, 368–370	skeletons. See skeletons
airplane. See toy airplane	solar system. See solar system animation
alien hands	summary, 329–330, 391
fingers, 168–169 , <i>168–170</i>	in work flow, 3
lattices for, 166–168 , <i>166–168</i>	Animation menu set, 41
aligning	Animation Preferences window, 303
camera, 459	Anisotropic shader, 222 , 222
images and UVs, 284, 285, 289–290, 290	ankle joints for block man
reference planes, 52, 52	heel control, 386
Alpha Channel (Mask) option, 437	hierarchies, 332, 333
alpha channels, 437–438, 443, 444	leg rigging, 362–364
Alpha Gain setting	skeleton, 338
displacement maps, 500, 502	anti-aliasing
fractals, 250, <i>250</i>	Maya Software rendering method, 439
reflection maps, 497, 497	mental ray rendering, 462–463
Alpha Is Luminance attribute, 497	Anti-aliasing Contrast setting, 463
Ambient Color attribute, 224 , 224	anticipation
Ambient lights, 400 , <i>400</i>	in animation, 8
ambient occlusion, 482–488 , <i>482</i>	in axe throwing, 308–310 , <i>310</i>
Ambient Shade settings, 400, 401	Applications category, 558
angles	Arc And Cone End Radius attribute, 404
camera shutter, 456	Area lights, 403 , <i>404</i>
normals, 75–76	armatures, 4
Spot lights, 402, 403	arms
sun light, 416–417, <i>416–417</i>	block man, 334
animation, 3, 295	control shapes, 383, 383
axe throwing. See throwing axe	skeleton, 337, 337–338
basic concepts, 6–8 , <i>7</i>	walk cycle, 341-346
bouncing ball. See bouncing ball	catapult
catapult	animating, 324–329 , <i>324–329</i> , 524–526, <i>525</i>
deformers, 325-329, 326-329	building, 139–141 , <i>139–141</i>
dynamics, 522-526, 523-526	IK, 362–363, 365
winch, 324–329 , 324–329	pontoons, 188–194, 189–193

pump, 375, 379 , <i>379</i>	
wheel, 377–382 , <i>377–378</i> , <i>380</i>	healt lights 207 209 207
Artisan module, 154	back lights, 397–398 , <i>397</i>
aspect ratio	back wheels rigging, 380–381 , <i>380–381</i>
camera setup, 459	background lights, 397
description, 5	baking simulations, 526–528 , <i>527–528</i>
film gate, 452	balls
reference images, 179	bouncing. See bouncing ball
resolution setting, 438	relationship with cones, 371, 371
assembling	base, catapult, 116
	Booleans, 120–121, 121–122
living room scene, 422–424 , <i>423–424</i>	ground spikes, 127–128 , <i>127–128</i>
toy airplane, 215–217, 216–217	Ngons, 122–123 , <i>122–123</i>
assets, 2	objects, 117–119 , <i>118–120</i>
Assign Material to Selection option, 25	winch baseboards, 124–126 , <i>124–126</i>
Attach to Motion Path dialog box, 317, 321, 321	baseboards for catapult base, 124–126 , <i>124–126</i>
Attaching Existing Cache File option, 557	basket assembly for catapult
attaching skeleton geometry, 339–341 , <i>339–341</i> , 352	animating, 326
attributes and Attribute Editor, 10–11, 11, 41	projectile shooting, 523–526, <i>523–526</i>
cameras, 451–455, 451–456	screw heads, 140, 140
creating, 372–373	Batch Render Animation dialog box, 457
emitters, 532–533 , 532–533	Batch Render Frame dialog box, 457, 457
flag, 556	batch rendering
lifespan, 534–535	process, 456–457 , <i>457</i>
lighting, 399 , 399–400	with render layers, 476 , 476–477
mental ray lighting, 414, 415	wine bottle, 461
nCloth, 552–554, <i>554</i>	bend deformers
nParticles, 534–537 , <i>534–537</i>	catapult animation, 325-329, 326-329
overview, 53 , 53, 76 , 76	working with, 162–163 , <i>163</i>
Paint Effects, 550–551	Bend Resistance attribute, 554
particles, 541–543 , <i>541–543</i>	bevel surfaces, 150 , <i>150</i>
Ramp shader, 223 , <i>223</i>	Bevel Plus surfaces, 150
raytraced shadows, 408	Bevel tool, 90–91 , <i>91</i> , 150
rendering, 544–547 , <i>544–546</i>	bevels
rigid bodies, 521–522	catapult base, 117-119, 118
shading and shaders	decorative box, 59–60, 60
common, 223–227 , <i>224–226</i>	edges, 70-71, 73
Mercury, 23, 23	Modeling Toolkit, 92–93 , 92–93
particles, 535–537 , <i>535–537</i>	toy airplane stabilizer, 199-200, 199
texture nodes, 235, 235	Bézier curves, 144, 144
Auto Keyframe feature, 41	Bézier splines, 313–314
axe throwing, 309, 311	bind poses, 340
bouncing ball, 298	Bind Skin option, 354
planets, 34	binding skeletons
Auto Load option, 461	feet to joints, 340
Automatic Mapping option, 275, 276	Heat Map method, 358–360 , <i>359</i>
Automatically Fit Bevel To Object option, 91	interactive skin binds, 360–361 , <i>360–361</i>
AVI movie format, 5	overview, 352–355 , <i>353–355</i>
axe model	smooth binds, 355–358 , <i>355–358</i>
deformers for, 163–165 , <i>164–165</i>	bitmapped textures, 232
throwing. See throwing axe	blades for toy airplane propeller, 205–206 , 205–206
axes, 5-6 , 6	Blinn shaders
axles for catapult wheels, 129, 129, 133-134, 134	description, 221 , 222
	1 '

living room scenes, 509-512, 511	bowl in catapult animation, 523, 524-526
in projections, 233	Bowling Lane scene, 522
red wagon, 241, 241	box modeling, 82
toy airplane, 227, 227, 229–231, 229–231	boxes, decorative. See decorative box
Blinn surfaces, 221	braces for catapult axles, 132, 132
block man	brackets for catapult
attaching elements to, 339-341, 339-341	axles, 129, 129, 134, 134
Forward Kinematics, 334 , <i>334</i>	winch, 124-125, 135
rigging, 382	Break Connection option, 239
control setup, 384–386 , <i>384–385</i>	Break Tangents icon, 314
control shapes, 382–383 , <i>383–384</i>	Bridge tool, 106 , <i>106–107</i>
heel controls, 386–390 , <i>386–390</i>	Brush Profile attribute, 550
leg, 362–363 , <i>362–363</i>	Brush Width attribute, 550
skeletons, 335–339 , <i>336–338</i>	Bucket tool, 260
walk cycle	Bulge texture, 237
Forward Kinematics, 341–348 , <i>341</i> , <i>343–345</i> ,	bump maps, 220
347	description, 225 , 225
Inverse Kinematics, 363–365 , <i>364–365</i>	red wagon wheels, 248–252 , 248–252
blocking, 297	buttons, mouse, 41
blocking out pontoons, 184–186 , <i>184–186</i>	By Frame attribute, 436
blowing grass and flowers, 547–549 , <i>548–550</i>	-
blur, motion, 456	
Blur by Frame attribute, 456	
Blur Length attribute, 456	caching
bodies	nCloth, 556–557 , <i>556</i>
dynamics. See dynamics and dynamics engine	nParticles, 537-538
human. See skeletons	Camera, Aim, and Up cameras, 449
red wagon	Camera and Aim cameras, 449
color, 241–243, 241–243	cameras, 448
decals, 253–254 , <i>253</i>	attributes, 451–455 , <i>451–456</i>
rigid, 518	creating, 448
active and passive, 518–521 , <i>519–520</i>	motion blur, 456
attributes, 521–522	in rendering, 437–438 , <i>437</i> , 459–460 , <i>460</i>
moving, 521	types, 449 , <i>449–450</i>
toy airplane. See toy airplane	Cancel Batch Render option, 457
bones, 332–333, <i>333</i>	candle holders, 154
Booleans, 120–121 , <i>121–122</i>	creating objects, 154–158 , <i>154–158</i>
bottle rendering, 458–460 , <i>459–460</i>	detailing objects, 158–161 , <i>159–160</i>
bounce in mental ray, 441	Caps Lock key, 46
Bounciness attribute, 522	cartoon hands, polygons for, 93–100 , 94–100
bouncing ball	carvings, decorative box
animating, 297–299	displacement mapping, 498–503, 500–503
creating, 296–297 , 296	reflections, 494–497, 497
Graph Editor for, 299–304 , <i>300–303</i>	case-sensitivity, 11, 559
hierarchy, 297 , 297	Casts Shadows option, 410, 420
playing back, 303	catapult, 115
rigid bodies for, 519–520 , <i>519</i>	animation
rotating, 306–307 , <i>306</i>	deformers for, 325–329 , <i>326–329</i>
squash and stretch, 305–306 , <i>305</i>	dynamics, 522–526 , <i>523–526</i>
timing for, 304–305 , <i>304</i>	winch in, 324–329 , <i>324–329</i>
boundary surfaces, 150 , <i>151</i>	arms
bounding box dollies, 12	animating, 324–329 , 326–329, 524–526, 525
bounding box donies, 12	building, 139–141 , <i>139–141</i>

base, 116	red wagon wheels, 243-246, 244-247
Booleans, 120–121 , <i>121–122</i>	shaders, 22–23
ground spikes, 127-128, 127-128	toon shading, 292–293
Ngons, 122–123 , <i>122–123</i>	toy airplane, 228
objects, 117–119, 118–120	volumetric lighting, 418
winch baseboards, 124–126 , <i>124–126</i>	Color attribute
production process, 116 , <i>116–117</i>	lighting, 399
wheels, 129–134 , <i>129–134</i>	nParticles, 536, 537
winch	shaders, 223
animating, 324–329 , <i>324–329</i>	Color Balance attributes, 235
assembly, 134–138 , <i>135–139</i>	Color Chooser window, 23, 24, 421, 544
baseboards, 124–126 , <i>124–126</i>	Color Range attribute, 404
caustics, 412, 441	Color1 and Color2 attributes, 550
CCIR 601/Quantel NTSC resolution, 438	Colors window, 560, 560
Center of Mass attribute, 522	Combine tool, 108–111 , <i>109–110</i>
Center Pivot option, 305–306	Command line, 71 , <i>71</i>
chains, IK, 365–366 , <i>366</i>	Common tab for rendering, 435
Channel Box, 10–11, 11, 41	Component Mode icon, 62
attributes, 76 , 76	compositing Ambient Occlusion results, 488 , 488–489
icons, 62 , 62	computer graphics (CG)
lighting, 431	3D coordinate space and axes, 5–6 , <i>6</i>
overview, 53	creation process, 2
reference planes, 51	production workflow, 2–4 , 3
sun settings, 17–18, 18	Cone Angle setting, 402
channels in rendering, 437–438 , <i>437</i>	cones
Character Set menu, 41	creating, 233–234 , <i>234</i>
Character Set menu, 41 Checker texture, 237	relationship with balls, 371, 371
child nodes, 29–30 , <i>29</i> , 36	Connect tool, 103–104, 104
Chord Height Ratio option, 152	Connection, 103–104, 104 Connection Editor, 375–376, 376
circles for block man, 383, 383	connections
clipping planes, 452	NURBS curves, 146
Closest Distance option, 353–358 , <i>356–358</i>	wheel control, 375–376, 376
	Constant attribute, 534–535
Closest On Hierarchy option, 354 Cloth texture	constraints, animation, 366–371 , 367–370
clothing, 237	construction history, 152
nCloth, 551	Control Points tessellation method, 153
caching, 556–557 , <i>556</i>	control vertices (CVs)
flag, 555–556 , <i>555–556</i>	attaching to skeletons, 352
tablecloth, 552–554 , <i>552–554</i>	glass candle holders, 154–158 , <i>154–158</i>
Cloud particle type	in NURBS, 144–146, 144–145
attributes, 535–536, 536	controls, block man
emitting, 531 , 532–533, 538–539, 539–540	setup, 384–386 , <i>384–385</i>
workflow, 546, 546	shapes, 382–383 , <i>383–384</i>
clusters in binding, 352 collisions	Controls attribute, 451
nCloth, 552	converting
,	NURBS to polygons, 153 , <i>153</i>
nParticles, 530, 543, 543	UVs to faces, 257, 257
rigid bodies, 518, 520–521	coordinates
color, 3	3D, 5-6, 6
customizing, 560, 560	UV, 254–269 , 254–269
decorative box, 282–290 , 282–291	copying
joints, 356–357	polygon faces, 112
ramp textures, 223	UVs, 263–264 , 284, 285

core concepts in computer graphics	decays, bounce, 303, 303
3D coordinate space and axes, 5–6 , 6	decorative box, 50 , <i>50</i> . <i>See also</i> living room scene
creation process, 2	displacement mapping, 493–494, 493
production workflow, 2–4 , <i>3</i>	process, 498–503 , 499–503
Cosine Power attribute, 227	reflection maps, 494–498 , 494–498
Count method for tessellation, 153	edges and faces, 67–68 , 68–70
Coverage setting, 235	editing, 63–65 , 63–66
Create Active Rigid Body option, 519	finishing, 71–75 , <i>72–75</i>
Create/Bins panel, 56 , <i>57</i>	modeling, 58–60 , 58–61
Create Deformers option, 166–167	photo-real mapping, 274
Create Emitter option, 531, 538	color, 282–290 , <i>282–291</i>
Create menu, 41	feet, 274–282 , <i>275–281</i>
Create nCache Options dialog box, 556, 556	reference planes
Create nCloth option, 555	creating, 50–52 , <i>51–52</i>
Create New Cache option, 538, 556	mapping, 53–56 , 54–56
Create Passive Collider option, 543, 552	Default Color attribute, 235
Create Passive Rigid Body option, 519	default renderer, 445-446
Create PSD Network option, 238, 238	deformations, 466
creation nodes, 28	deformers, 161
cross braces for catapult base, 118, 119	axe model, 163–165 , <i>164–165</i>
cubes	bend, 162–163 , <i>163</i>
decorative box, 58–60 , 58–61	catapult animation, 325–329 , <i>326–329</i>
toy airplane body, 194–195, <i>195</i>	flare, 164, <i>164</i>
toy airplane wings, 207, 207	history for, 165
Current Time indicator, 70	lattices. See lattices
curvature with bend deformers, 163, 326–328 , 327–329	Soft Modification tool, 161-162, 162
curves	wave, 369
animation, 528–529 , <i>529–530</i>	Delete Cache option, 538, 556
axe throwing, 309–315 , <i>310</i>	deleting history, 72, 152
flying text, 320–322 , <i>320–323</i>	denormalizing views, 310
in Graph Editor, 300–302 , <i>301–302</i>	Density attribute
NURBS. See NURBS (Non-Uniform Rational	nParticles, 546
B-Splines) modeling and surfaces	volumetric lights, 418
organic, 144 , <i>144</i>	depth in space, 5
path animation, 317	Depth Map shadows, 407
customizing	depth of field (DOF), 515, 515
interface, 557–561 , <i>557–558</i> , <i>560–561</i>	Detach Skin option, 355
shelf, 558–559 , <i>558</i>	detaching skeletons, 355
Cut Faces tool, 112 , <i>112</i>	details
CV curves for path animation, 317	catapult wheels, 131, 131
CVs (control vertices)	glass candle holders, 158-161, 159-160
attaching to skeletons, 352	in models, 80, 80
glass candle holders, 154-158, 154-158	NURBS, 145 , <i>146</i>
in NURBS, 144-146, 144-145	poly meshes, 101 , <i>101</i>
cylinders, bend deformers for, 162-163, 163	toy airplane engine, 203-204 , <i>203-204</i>
	Device Aspect Ratio attribute, 438
	Dice scene, 522
D	Difference operator, 121, 121
DAG nodes, 29	Diffuse attribute, 225 , 225
Damping attribute, 522	diffuse lighting, 399, 400
decals, 253–254 , <i>253</i>	diffusion, 220
Decay Rate attribute	digital images, 4
Point lights, 401, 402	formats, 5
Spot lights, 402	resolution, aspect ratio, and frame rate, 5

ext format, 435 extensions, filename, 435 Extract tool, 112, 112 Extrude option, 85–86 extrusions and Extrude tool bevels, 92–93, 92–93 catapults, 124–125, 124–125 faces decorative box, 64–65, 65 polygons, 85–86, 85–86 pontoon arms, 193 surfaces, 147–149, 148 tools Modeling Toolkit, 86–89, 87–90 poly, 85–86, 85–86 toy airplane wings, 208	film gate, 459–460 Film Gate attribute, 452 filtering for mental ray rendering, 462–463 Final Gather (FG) image-based lighting, 413 noise, 513–514, 513–514 process, 478–481, 479–481 fingers creating, 93–100, 94–100 lattices for, 168–169, 168–170 set driven keys, 372–373, 374 Fit Resolution Gate attribute, 452 fixed references, 5 FK (Forward Kinematics), 332, 333 block man, 334, 334 walk cycle, 341–348, 341, 343–345, 347 flag, 555–556, 555–556 flare deformers, 164, 164
F	Flat Tangents icon, 314
faces	floating menus, 42 , 42
converting UVs to, 257, 257 decorative box, 67–68 , 68–70	floor for red wagon, 270 , 270
extruded	Flow Path Object Options window, 321
decorative box, 64–65, 65	flowers and grass, 547–549 , <i>548–550</i>
polygons, 85–86 , <i>85–86</i>	fluid dynamics, 518
polygon, 81, 81, 90–91 , 91	flying text, 320–322 , <i>320–323</i>
toy airplane wings, 210, 210	Focal length attribute, 451–452 , 452
Falloff Radius attribute, 49	focal length of cameras, 451–452 , <i>452</i>
Far Clip Plane attribute, 452	Fog Intensity setting, 418
FCheck frame player, 5, 461, 461	Fog Spread setting, 418, 418 Follow feature, 317
feathering, 287–288, 288	follow-through
feet	_
block man	animation, 8
control shapes, 383, 384	axe throwing, 312–315 , <i>313–315</i> Forward Kinematics (FK), 332 , <i>333</i>
heel control, 386–390 , <i>386–390</i>	block man, 334 , 334
joints, 340	walk cycle, 341–348 , <i>341</i> , <i>343–345</i> , <i>347</i>
decorative box	fps (frames per second), 5
creating, 63–64 , <i>64</i> , <i>66</i>	Fractal texture, 236
photo-real mapping, 274–282 , <i>275–281</i>	Fractional Tolerance option
FG (Final Gather)	NURBS from polygons, 153
image-based lighting, 413	polygons from NURBS surfaces, 152
noise, 513–514 , <i>513–514</i>	Frame All option, 45
process, 478–481 , <i>479–481</i>	Frame/Animation Ext list, 436
fields of blowing grass and flowers, 547–549 , <i>548–550</i>	frame numbers in filenames, 435
fields with dynamic bodies, 518	Frame Padding attribute, 436, 436
file formats, 5	frame ranges in rendering, 436 , 436
File menu, 41	frame rate, 5
file name prefix setting, 435	Frame Selection option, 45
file nodes for textures, 237–239 , <i>237–240</i>	frames, 3, 7
filenames, 435	frames per second (fps), 5
fill lights, 396–397 , <i>397</i>	framing, 6
Fill Options option, 443	Freeform Layout option, 334, 335
Film Back attribute, 452	, , ,

creation process, 2

production workflow, 2-4, 3

Height Ratio attribute, 20

Height setting in rendering setup, 438

HDRI, 490-493, 490-492 living room scene rendering, 504-513, 505-513

toy airplane body, 195, 195

stabilizer, 198, 198

hands, 348-350, 349-351

Intensity attribute, 399	keyframes for, 341-348
Interactive Creation feature, 17, 17	moving, 350, 351
Interactive Creation option reference planes, 51, 51	rotating, 333, 333–334
Interactive Photorealistic Renderer (IPR), 445, 445	skeletons, 332-333, 333-334
Interactive Skin Bind, 360–361 , <i>360–361</i>	JPEG (Joint Photographic Experts Group) format, 5,
Interactive Split tool, 63, 63, 67–68, 68	435, 437
interface, 10, 39	
Attribute Editor, 76 , <i>76</i>	
Command line, 71, 71	K
customizing, 557–561 , <i>557–558</i> , <i>560–561</i>	Keep Faces Together option, 89, 89
decorative box. See decorative box	Keep Image option, 444
Help line, 71–72 , <i>71</i>	Keep Image Width/Height Ratio option, 257
Hypershade, 56 , <i>57</i>	Keep Unbaked Keys option, 526
Layer Editor, 57–58 , <i>57–58</i>	key lights, 395–396 , <i>395–396</i>
manipulators, 46-47, 46	Key Selected option, 431
marking menus, 42	keyboard shortcuts
Modeling Toolkit, 77, 77	capital letters with, 46
navigating, 40	customizing, 559–560 , <i>560</i>
Outliner, 76–77 , <i>77</i>	overview, 12
perspective/orthographic panels, 43-45, 44-45	for viewing, 43
Range Slider, 70–71 , <i>70</i>	keyframes, 6 , <i>7</i> , 297–298
screen components, 10–11 , <i>11</i> , 40–42 , <i>40–42</i>	axe throwing, 313–315
Shelf, 67	bouncing ball animation. See bouncing ball
Soft Selection, 47–49 , 47–48	joints, 341–348
Status line, 61–63 , <i>61</i>	moving, 302–303, 303
summary, 78	for timing, 304
symmetry, 49 , 49	walk cycle, 341-348 , <i>341</i> , <i>343-345</i> , <i>347</i>
Time Slider, 70–71 , <i>70</i>	kinematics, 332
Tool Box, 67 , 67	FK, 332-333
interpenetration, 172	block man, 334, 334
interpolation	walk cycle, 341-348 , <i>341</i> , <i>343-345</i> , <i>347</i>
ramp texture, 236	IK, 332–333, 362
in timing, 304	rigging legs, 362–363 , <i>362–363</i>
Inverse Kinematics (IK), 332–333, 362	walk cycle, 363–365 , <i>364–365</i>
block man, 382-383, 383-384	knee joints
rigging legs, 362–363 , <i>362–363</i>	rigging, 362
walk cycle, 363–365 , <i>364–365</i>	skeletons, 332, 333, 338
IPR (Interactive Photorealistic Renderer), 445, 445	knuckles
isoparms, 144, 144	alien, 168, <i>169</i>
	creating, 99–100 , <i>99–100</i>
	Heat Map method, 359
J	interactive skin bind, 361
jars, 154	set driven keys, 373–374
creating objects, 154–158 , <i>154–158</i>	
detailing objects, 158-161, 159-160	
Joint Photographic Experts Groups (JPEG) format, 5,	L
435, 437	Lambert shaders
joints and Joint tool, 336	creating, 54–55, <i>55</i>
attaching, 339-341, 339-341	diffusion with, 221 , 221
in binding, 352–355 , <i>353–355</i>	reference images, 180–181
block man, 383, 386-387	solar system animation, 23
creating, 336–338, 336–338	lattices, 166

animation through, **170–172**, *171–173*

fingers, 168–169 , <i>168–170</i>	IBL
flying text, 322 , <i>323</i>	description, 413
hands, 166–168 , <i>166–168</i>	HDRI, 490–493 , 490–492
toy airplane, 216, <i>216</i>	living room scene rendering, 504–513 , <i>505–513</i>
Layer Editor, 41	light types, 400–405 , 400–404
icons, 62 , 62	linking, 405–407 , 405–406
overview, 57–58 , <i>57–58</i>	living room scene, 424–425 , <i>425</i>
Layered shader, 222-223, 222	manipulators, 431–432 , <i>432</i>
layers	mental ray
block man, 383, 383	global illumination, 412–413, 413
creating, 183-184, 184	image-based, 413
render, 467–468	physical sun and sky, 414–417 , <i>414–417</i>
batch rendering for, 476 , 476–477	need for, 394 , 394–395
mental ray, 468	practice for, 430
overrides, 473–476 , <i>474–476</i>	in shading, 220
still life, 468–471 , 469–471	shadows, 394, 407–412
testing, 471–473 , <i>472–473</i>	summary, 432
leather simulation, 554, 554	three-point, 395–398 , <i>395–397</i>
Legacy Sampling mode, 463	linking light, 399, 405–407 , 405–406
legs	Live Forever attribute, 534
block man, 383, 383	living room scene
moving. See kinematics	assembling, 422–424 , 423–424
rigging, 362–363 , <i>362–363</i>	glass, 427–429 , 428–429
lemon, rendering, 468–471 , 469–471	lights, 424–425 , <i>425</i>
batch rendering for, 476 , 476–477	reflection and shadows, 425–427 , <i>426–427</i>
layer overrides, 473–476 , <i>474–476</i>	rendering, 482–488 , 482, 504
testing, 471–473 , 472–473	depth of field, 515 , <i>515</i>
Length Min/Max attribute, 551	IBL, 504–513 , <i>505–513</i>
lens flare, 419–420 , <i>419–420</i>	noise, 513–514 , <i>513–514</i>
Lens Flare attribute, 420	wrapping up, 516
Lifespan Mode attribute, 541	Loaded option for mental ray rendering, 461
lifespan of nParticles, 534–535	loading images, 444
Lifespan Random attribute, 541	Local Axis, 6, 309
LifespanPP Only attribute, 535	Local Z attribute, 94
Light Angle setting, 411	locking cameras, 460
Light Fog attribute, 418	locomotive
light glow effect, 419–420 , <i>419</i>	animating, 538–547 , <i>539–546</i>
Light Linking window, 521	pumps, 375, 379 , 379
light probes, 490, 490	rigging. See rigging
Light Radius setting, 411, 411, 427	lofting method, 146–147 , <i>146</i>
Light Shape attribute, 404	logos, flying, 320–322 , <i>320–323</i>
Lighted mode, 45, 45	Look at Selection option, 45
lightFog node, 418	Low Bound attribute, 163, 326–327
lighting, 4, 393	low-count poly models, 81
animating, 431	ion count poly models, or
attributes, 399 , <i>399–400</i>	
concepts, 394	M
effects, 418	Macintosh keys, 12, 43
lens flare, 419–420 , <i>419–420</i>	main menu Bar, 10
shader glows, 420–421 , <i>421–422</i>	Main Menu bar, 11, 41–42 , 41–42
volumetric lighting, 418–419 , <i>418</i>	manipulators
Final Gather, 478–479	lights, 431–432 , 432

overview, 46-47, 46

pivot point, 32, 32	physical sun and sky, 414-417 , <i>414-417</i>
projections, 234, 234	settings, 461–467 , <i>461–462</i>
maps and mapping	menu sets, 26, 41–42 , 41–42
bump, 220	menus, 41–42 , 41–42
description, 225 , 225	floating, 42 , 42
red wagon wheels, 248-252, 248-252	marking, 42 , <i>43</i>
decorative box	submenus, 42, 42
color, 282–290 , <i>282–291</i>	Mercury
displacement, 493-494, 493	animating, 30–31 , <i>30</i>
feet, 274–282 , <i>275–281</i>	creating, 18–19 , <i>19</i>
photo-real, 274	grouping, 28
color, 282–290 , <i>282–291</i>	rotation of, 27–28
feet, 274–282 , <i>275–281</i>	Merge Component tool, 108-111, 109-110
reference planes	Merge Vertex tool, 108-111, 109-110
Hypershade, 53–56 , <i>54–56</i>	mesh surface holes, 113-114, 113-114
pontoons, 179-184, 180-184	mia shaders, 512, <i>512</i>
reflection, 439, 494-498, 494-498	Min Distance setting for emitters, 533, 533, 540
texturing and textures, 220	Min Sample Level setting, 462–464
UV, 232	Mirror Geometry option, 125, 126
marbles scene, 522	Mirror U setting, 235
marking menus, 42, 43	Mirror V setting, 235
mass	Modeling Toolkit tools, 77, 77
in animation, 7	Bevel, 92-93 , 92-93
of rigid bodies, 521	Bridge, 106 , <i>106–107</i>
Mass attribute, 521	Connect, 103-104, 104
matchmoving, 455	drag selection, 104–106 , <i>104</i>
material nodes, 220	Extrude, 86–89 , 87–90, 124–125 , 124–125
materials	Multi-Cut, 102, 102
red wagon wheels, 248-252, 248-252	poly editing, 84-85
solar system animation, 23	Symmetry selection, 107–108 , <i>107–108</i>
Matte Opacity Mode and slider, 226	toy airplane
mattes, 227 , 437	body, 197
Max Distance setting for emitters, 533, 533, 540	pontoon arms, 188–194, 189–192
Max Influences setting, 354	stabilizer, 199–200
Max Sample Level setting, 462-464	wings, 208, 210
Maya Embedded Language (MEL), 71 , 71, 558-559	models
Maya Hardware rendering method, 440 , 440	with NURBS. See NURBS (Non-Uniform Rational
Maya Software rendering method, 434, 438–439 , 440	B-Splines) modeling and surfaces
Maya Vector tab, 442, 442	planning, 80 , <i>80</i>
MEL (Maya Embedded Language), 71 , 71 , 558–559	with polygons. See polygons
memory caching of particles, 537	Modify menu, 41
mental ray rendering, 440-441, 504	momentum
attributes, 227	axe throwing, 315
Final Gather	rigid bodies, 521
depth of field, 515 , <i>515</i>	moons
noise, 513–514 , <i>513–514</i>	creating, 22 , 22
process, 478–481 , <i>479–481</i>	Earth, 32–34 , <i>32</i>
wrapping up, 516	motion
IBL, 504–513 , <i>505–513</i>	animation. See animation
image-based, 413	simulating. See dynamics and dynamics engine
layers, 468	motion blur, 456 , 465–467 , <i>465–467</i>
lighting, 412–413 , 413	motion paths, 321, 321

Mountain texture, 236 mouse controls, 11–13, 41 Move tool for hands, 350 working with, 46, 46 Move And Sew The Selected Edges tool, 278, 278, 281 Move Pivot tool, 350 moving joints, 350, 351 keyframes, 302–303, 303 objects, 46, 46 rigid bodies, 521 Multi-Cut tool, 99, 99, 102, 102 multiple versions, saving, 21	setting, 541 – 543 , <i>541</i> – <i>543</i> shading, 535 – 537 , <i>535</i> – <i>537</i> caching, 537 – 538 emitting overview, 530 – 533 , <i>531</i> – <i>533</i> steam, 538 – 540 , <i>539</i> – <i>540</i> NTSC (National Television Standards Committee) resolutions, 438 null nodes, 36 Number of Samples setting, 440, 463 NURBS (Non-Uniform Rational B-Splines) modeling and surfaces, 143 , <i>144</i> Artisan module, 154 bevel, 150 , <i>150</i> boundary, 150 , <i>151</i> cones, 233 – 234 , <i>234</i>
names nodes, 23	construction history, 152 converting to polygons, 153 , <i>153</i> creating, 46
objects, 15 , 17–18, <i>18</i> National Television Standards Committee (NTSC) resolutions, 438	details, 145 , <i>146</i> extruded, 147–149 , <i>148</i> glass candle holders, 154
navigating, 40 nCloth dynamics, 551 caching, 556–557, 556 flag, 555–556, 555–556 tablecloth, 552–554, 552–554 nDynamics menu set, 41 Near Clip Plane attribute, 452 New Scene option, 17 Ngons for catapult base, 122–123, 122–123 Node Editor for block man, 384–386, 385 nodes inherited, 36 objects, 28–29 renaming, 23	creating objects, 154–158 , <i>154–158</i> detailing objects, 158–161 , <i>159–160</i> lofting method, 146–147 , <i>146</i> organic curves, 144 , <i>144</i> overview, 144–146 , <i>144–145</i> planar, 149–150 , <i>149</i> polygons, 151–152 revolved, 147 , <i>147</i> spheres creating, 234 , <i>234</i> lattices for, 170–172 , <i>171–173</i> summary, <i>174</i> techniques, 146–147
shader, 220 textures, 235–239, 235–240 noise Final Gather, 513–514, 513–514 toy airplane, 228–229, 228–229 Noise attribute, 236 Noise Frequency attribute, 236 Noise texture, 236 Non-Uniform Rational B-Splines. See NURBS (Non-Uniform Rational B-Splines) modeling and surfaces normal constraints, 369–370, 370 normalized skin weights, 357 normalizing views, 310, 310 Normals, 75–76 nParticles, 530 attributes, 534, 534 lifespan, 534–535	Object Mode for pontoons, 185, 186 Object Mode icon, 62 objects, 28 catapult base, 117–119, 118–120 glass candle holders creating, 154–158, 154–158 detailing, 158–161, 159–160 grouping, 36 hiding, 72, 72 hierarchies, 29–30, 29, 315, 316 manipulating, 13–14, 13–14 names, 15, 17–18, 18 nodes, 28–29 replacing, 318–319, 319 occlusion, ambient, 482–488, 482

Particle Render Type menu, 535, 535

Offset Edge Loop tool	particles and particle dynamics, 518
poly surfaces, 103 , <i>103</i>	animating, 538–547 , <i>539–546</i>
pontoon arms, 191, <i>191</i>	nParticles. See nParticles
Offset values	Pass Contribution Maps feature, 468
aim constraints, 369	passive rigid bodies, 518–521 , <i>519–520</i>
orient constraints, 367	pasting for UVs, 284, 285
Omni emitters, 531–532, 532	patches, NURBS, 144
opacity	paths in axe throwing, 317–318 , <i>317</i>
matte, 226	pelvis
nParticles, 544-545, 544	block man, 334
UV layouts, 260, 260	control shapes, 383, 384
Opacity Scale attribute, 536, 537, 544, 544	leg rigging, 363–365
Open Image option, 444	skeleton, 337, 338, 339
Open Render Settings window, 443, 444	Inverse Kinematics for, 363–365
OpticalFX node, 419	for root, 335
option boxes, 42, 42	Penumbra attribute, 404
Option key, 12	Penumbra Angle attribute
organic curves, 144, 144	shader glows, 420
orient constraints, 367, 367	spot lights, 402, 403
origins	perception of mass, 7
3D coordinates, 5	persistence of vision, 3
UV coordinates, 254	perspective shifts in images, 178
orthographic views, 43-45, 44	perspective view, 43–45 , <i>44</i> , 448
Out Color attribute, 484	Phong E shaders, 221, 222
Outliner	Phong E surfaces, 221
hiding objects, 72, 72	Phong shading, 221 , <i>221</i>
interface, 76–77 , <i>77</i>	photo-real mapping decorative box, 274
objects in, 29, 29	color, 282–290 , <i>282–291</i>
solar system animation, 34–36 , <i>34</i> , <i>36</i>	feet, 274–282 , 275–281
outlines for toon shading, 292–293, 293	Photoshop
outputting solar system animation, 37, 37–38	decorative box color, 287–290 , 288–291
overlighting, 394, 394	files, 238–239 , 238–240
overrides in render layers, 473–476 , <i>474–476</i>	UV coordinates in, 259-260, 259-261
Overscan setting, 453, 453	Physical Sun and Sky (PSAS) lighting, 414-417,
	414-417
	physics in motion simulation, 518
Р	pick walking, 337
padding frames, 436, 436	pivots and pivot points
Paint Effects, 517	block man, 334
attributes, 550-551	center of balance as, 26
toon shading, 291–293 , <i>292–293</i>	Earth and moon, 32, 32
working with, 547–549 , <i>548–550</i>	joints, 332, 333
Paint Effects to Polygons option, 551	Mercury, 28
Paint Skin Weights tool, 356–359, 356	Pixel Aspect Ratio attribute, 438
painting skin weights, 356–358 , <i>356–358</i>	Place2dTexture nodes, 235–236, 235
panels	planar surfaces, 149–150 , <i>149</i>
perspective/orthographic, 43-45, 44-45	planets
red wagon model, 254–269 , <i>254–269</i>	Earth, 32–34 , <i>32</i>
screen, 10, 11, 40–41	grouping, 28, 33
parallax in images, 178	Mercury, 18–19 , <i>19</i>
parent constraints, 371	Pluto, 35–36 , <i>36</i>
parent nodes, 29–30 , 29, 36	rotation, 27–28

Saturn, 20 , 21	Slide Edge, 111 , <i>111</i>
Venus, 31-32	Smooth, 113
planning models, 80, 80	Split Mesh with Projected Curve, 113-114,
platforms for catapult base, 119, 119	113–114
Play Every Frame option, 519, 531	Wedge Face, 89–90 , 90
playblasting, 37 , <i>37</i> – <i>38</i>	Polygons menu set, 41
playing back	pontoons for toy airplane
axe throwing animation, 312-313	arms, 188–194 , <i>189–193</i>
bouncing ball, 303	in assembly, 215-216, 215-216
wine bottle rendering, 461	blocking out, 184–186 , <i>184–186</i>
Plug-in Manager	reference images, 178-184, 178-184
mental ray, 412, 461, 461	shaping, 186–188 , <i>186–188</i>
Modeling Toolkit, 87, 87	pose animation, 341
Pluto, 35–36 , 36	post processes, Paint Effects for, 551
point constraints, 366-367, 367	postproduction stage, 2
Point Density attribute, 481	Preferences window, 26, 27, 557–558 , 557
Point Interpolation attribute, 481	default renderer, 445
Point lights, 400-401	default settings, 40
point on poly constraints	prefixes, filename, 435
catapult, 524	preproduction stage, 2
description, 368, 368, 371	Preserve Aspect Ratio option
Poke tool, 90 , 90, 100, 100	decorative box, 51
pole vector constraints, 371	reference images, 179
Poly_Hand project, 93–100 , <i>94–100</i>	previewing rendering, 443-444, 443
poly meshes, details on, 101, 101	Primary Diffuse Scale setting, 481
Polygon tool, 83–84 , <i>83</i>	Primary Visibility option, 479
polygons, 79	primitives, 17
basics, 82	polygons, 83
catapult. See catapult	working with, 81 , 82
components, 81, 81	procedural shading, 253
converting NURBS to, 153, 153	procedural textures, 232
for hand, 93–100 , 94–100	production stage, 2
modeling method, 80	production workflow, 2–4 , <i>3</i>
NURBS surfaces for, 151–152	profile curves
poly meshes, 101 , <i>101</i>	catapult
primitives, 83	wheels, 130-131, 130-131
splitting, 63, 64	winch assembly, 134, 135
summary, 142	extruded surfaces, 147-148, 148
tools for	revolved surfaces, 147, 147
Add Divisions, 101 , <i>101</i>	toy airplane engine, 201, 202
Bevel, 90–91 , <i>91</i>	project folders, 15
Combine and Merge, 108-111, 109-110	Project Window, 15–16, 16
Cut Faces, 112, 112	projections on surfaces, 232-235, 233-234
Duplicate Face, 112	projects
Extract, 112, 112	creating, 15-16 , <i>16</i>
extrusion, 85–86 , 85–86	saving, 21
Insert Edge Loop, 102-103, 102	propeller blades, 205–206 , 205–206
Modeling Toolkit. See Modeling Toolkit tools	proxy objects, 318
Offset Edge Loop, 103, 103	PSD file node, 238–239 , 238–240
Poke, 90 , 90	PTGui program, 492-493, 492
Polygon, 83-84 , 83	pulleys for catapult winch assembly, 135-137, 135
Sculpt Geometry, 114-115, 115	pumps for steam locomotive arm, 375, 379, 379

Q	side panels UV coordinates, 254–269 , <i>254–269</i>
quads, 81	texturing, 240 , <i>240</i>
quality of rendering	shader assignment, 241-243 , <i>241-243</i>
anti-aliasing, 439 , 440	wheels, 243–252 , 243–252
sampling, 462–463 , <i>464</i>	toon shading, 291–293 , 292–293
steam, 544	Redo Previous Render option, 443, 443
Queue setting, 27	Reference dialog box, 423
QuickTime movie format, 5	reference images for pontoons, 178–184, 178–184
Quekrime movie format, 5	reference planes
	creating, 50–52 , <i>51–52</i>
R	mapping, 53–56 , <i>54–56</i>
Radial Frequency attribute, 419	Reflected Color attribute, 226–227
Radius attribute	reflection maps, 439, 494-498 , 494-498
particles, 541	reflections, 220
torus, 20	living room scene, 425-427 , 426-427
Radius Scale attribute, 537, 541, 541-542	in rendering, 439 , 446–447 , <i>446</i>
railings for red wagon	with shaders, 226-227
lighting, 414, <i>415</i>	Reflections attribute, 446-447, 446
shading, 270–273 , <i>271–273</i>	Reflectivity attribute, 226
Ramp shader, 223 , <i>223</i>	refractions in rendering, 439 , 447–448 , <i>447</i>
ramp textures, 223	refractive index, 447–448, 447
color transitions, 236 , 236	relationship editors for light linking
red wagon wheels, 243-246, 244-247	Light Linking window, 521
random noise patterns, 236	Relationship Editor, 405-406, 406
random particle speed, 533, 540	relationships in animation
Random Range attribute, 535	constraints, 366–371 , <i>367–370</i>
Range End Frame setting, 26	set driven keys, 371–374 , <i>371–374</i>
Range Slider, 26, 26, 41, 70–71 , 70	Remove Image option, 444
Range Start Frame setting, 26	renaming nodes, 23
ranges, frame, 436 , <i>436</i>	render node display area, 56 , <i>57</i>
raster images, 3	Render Passes feature, 468
Rate attribute	Render Region option, 443, 443
emitters, 533	Render Settings window
nParticles, 545	attributes, 544
Raytrace Options attributes, 226	batch rendering, 457
raytracing, 407	final gathering, 479
Maya Software rendering method, 439	mental ray for Maya, 461, 462
for reflections, 446–447 , 446–447	mental ray lighting, 412
with shaders, 226	reflections, 446–447
for shadows, 408–409 , 409–410	resolution, 438
reading animation curves, 300–302 , <i>301–302</i>	settings, 457, 458
rear stabilizers for toy airplane, 198–201, 198–201	tabs in, 434–435, 434
Receive Shadows option, 410	vector rendering, 442, 443
red wagon	wine bottle, 458
decals, 253–254 , <i>253</i>	Render The Current Frame option, 60, 61
finishing, 273–274 , <i>273–274</i>	Render View window
floor texturing, 270 , <i>270</i>	reflection maps, 497
lighting, 414–417 , 414–417	settings, 443–444 , <i>443</i>
railings	rendering, 2, 433
lighting, 414, 415	Ambient Occlusion, 482–488 , 482
shading, 270–273 , <i>271–273</i>	attributes, 535–536, 536–537, 544–547 , 544–546
rendering settings, 464–465 , 464–465	batch, 456–457 , <i>457</i> , 461
1011401111g octilings, 101-103, 101-103	outch, 150 157, 157, TOI

S

cameras in. See cameras	control shapes, 382–383 , <i>383–384</i>
decorative box, 60, 61	heel controls, 386–390 , <i>386–390</i>
default renderer, 445-446	skeletons
engine selection, 434–435	hands, 348-351, 349-352
Maya Hardware, 440 , <i>440</i>	legs, 362–363 , <i>362–363</i>
Maya Software, 439 , 440	steam locomotive, 375, 375
Maya Vector rendering method, 441–443, 443	back wheel, 380-381 , <i>380-381</i>
mental ray for Maya, 440-441	finishing, 382 , 382
filenames for, 435	pump arm, 379 , <i>379</i>
final gathering process, 478–481 , <i>479–481</i>	wheel arms, 377–378 , <i>377–378</i>
frame ranges, 436 , 436	wheel control, 375–376, 376
HDRI, 490–493 , 490–492	rigid bodies, 518
image channels, 437–438 , <i>437</i>	active and passive, 518–521 , <i>519–520</i>
image formats, 435-436	attributes, 521–522
IPR, 445 , 445	moving, 521
lighting, 399, 400	Rigidity attribute, 554
living room scene, 504	rim lights, 397–398
depth of field, 515 , <i>515</i>	rings for Saturn, 20 , 21
IBL, 504–513 , <i>505–513</i>	root joints
noise, 513–514 , <i>513–514</i>	description, 332–333, <i>333</i>
wrapping up, 516	pelvis for, 335
mental ray for Maya, 461–467 , 461–462	Rotate Frame setting, 235–236
motion blur, 456	Rotate tool, 46
previewing, 443–444 , <i>443</i>	rotating and Rotate attributes
red wagon, 464–465 , 464–465	axe throwing, 311–314, 314
reflections, 446–447 , <i>446–447</i>	baking out simulations, 526, 528
refractions, 446, 447–448	bouncing ball, 306–307 , <i>306</i>
render layers, 467–468	constraints, 367
batch rendering for, 476 , 476–477	joints, 333, 333–334
mental ray, 468	keyframe, 299, 311–314, <i>314</i>
overrides, 473–476 , 474–476	lens flare, 419
still life, 468–471 , 469–471	Local Axis, 6
testing, 471–473, 472–473	nParticles, 534
resolution, 438	planets, 27–28
saving and loading images, 444–445	set driven keys, 372
setup, 434–435 , <i>434</i>	wheel control, 376
solar system animation, 465–467 , 465–467	Rotation attributes, 367, 419
summary, 516	Roughness attribute, 227
wine bottle, 458–460 , <i>459–460</i>	roundness for bevels, 91
Rendering menu set, 41	Touriences for bevels, 71
Repeat UV setting, 235	
replacing objects, 318–319, 319	
resolution, 5	S Divisions attribute, 167
in rendering, 438 , 459	Sample Lock setting, 463
shader glows, 420	Sampling Mode, 462–463
volumetric lighting, 419	Saturn, 20 , 21
revolved surfaces, 147, 147	saving
RGB color, 224	images, 444–445
rigging	multiple versions, 21
block man, 382	projects, 21
control setup, 384–386 , <i>384–385</i>	Scale channels, 18
control setup, 301 300, 301 303	* */ *

Scale keyframe, 299	shader glows, 420–421 , <i>421–422</i>
scale manipulator, 18	shading and shaders, 3, 219-220
Scale tool, 14, 46	attributes
scaling and Scale attributes	common, 223–227 , 224–226
bouncing ball, 299, 305	nParticles, 535-537, 535-537
constraints, 371	nParticles, 546, 546
displacement maps, 502	Paint Effects, 550
nParticles, 534	red wagon, 241-243, 241-243
sun, 18, <i>18</i>	finishing, 273–274 , 273–274
Scene End Frame setting, 26	panels, 254–269 , 254–269
scene file icons, 62, 62	railings, 270–273 , <i>271–273</i>
Scene Start Frame setting, 26	wheels, 243–252 , <i>243–252</i>
scenes	reference images, 179-181, 180-181
living room. See living room scene	solar system animation, 22–25, 23–25
saving, 21	toon, 291–293 , 292–293
Scenes folder, 15	toy airplane, 208, 227-231, 227-231
screen components, 10–11 , <i>11</i> , 40–42 , <i>40–42</i>	types, 221–223
screw heads, 121–123, 121–123	Shading attribute, 550
Script Editor, 71, 71, 559	shadows, 407 , 407
scroll wheel, 12, 43	creating, 407–409
scrubbing Timeline, 27	lighting for, 394
Sculpt Geometry tool, 114–115, 115	living room scene, 425–427 , 426–427
SDKs (set driven keys), 371	noise, 514, <i>514</i>
creating, 371–372 , <i>371</i>	per object, 410
hands, 372–373 , <i>373–374</i>	raytracing, 411–412 , <i>411–412</i>
heel, 388–390 , <i>389–391</i>	render layers, 473–476 , <i>474–476</i>
Secondary Diffuse Scale setting, 481	wine bottle rendering, 458–459, 459
secondary motion	shape nodes, 28
animation, 313	shapes
axe throwing, 315-317, 316	block man, 382–383 , <i>383–384</i>
sections for NURBS, 146	primitives, 81 , 82
segments for bevels, 91	toy airplane
Select Camera option, 449	body, 194–197 , <i>194–197</i>
Select tool, 64, 64	pontoons, 186–188 , <i>186–188</i>
Select Edge Loop Tool, 74	Shelf, 10, 11, 40
Selected Color attribute, 544	decorative box, 63–65 , <i>63–66</i>
selection handles	overview, 67
for axe, 308, 319	Shelf Editor, 558–559 , 558
in rigging, 324	shortcuts, keyboard
selection modes, 62 , 62	capital letters with, 46
selections, 13	customizing, 559–560 , <i>560</i>
Separate option, 111	overview, 12
sequences in image planes, 455, 456	for viewing, 43
set driven keys (SDKs), 371	Show Batch Render option, 457
creating, 371–372 , <i>371</i>	Show Manipulator tool, 234, 431–432
hands, 372–373 , <i>373–374</i>	Shutter Angle attribute, 456
heel, 388–390 , <i>389–391</i>	sibling nodes, 36
Set Normal Angle window, 75, 75	side panels of red wagon model, 254–269 , 254–269
Settings category, 558	silk simulation, 553, 554
Settings tab, 26, 27	simplifying animation curves, 528–529 , 529–530
Shaded Brightness Two Tone option, 291	simulations
shaded mode, 45, 45	animation curves, 528–529 , <i>529–530</i>

surfaces, 83

Subdivisions Height setting, 163, 163

spanning surfaces, 144, 146

Sparse Curve Bake option, 526, 528

toy airplane, **227–231**, *227–231*

submenus, 42, 42	types of, 232
sun	UV mapping, 232
creating, 17–18 , <i>18</i>	in work flow, 3
mental ray rendering, 414–417 , 414–417	three-button mouse, 41
surface normals, 369	three-node cameras, 449, 450
Surface Shading attribute, 536	three-point lighting, 395–398 , <i>395–397</i>
surfaces	3D delta setting, 152
catapult wheels, 131, 131	3D objects and space, 5-6, 6
holes, 113-114, 113-114	blur process, 456
NURBS. See NURBS (Non-Uniform Rational	polygons for. See polygons
B-Splines) modeling and surfaces	textures, 239
projections for, 232–235 , 233–234	32-bit image files, 4
revolved, 147 , <i>147</i>	Threshold attribute, 536
subdivision, 83	throwing axe, 307
texture nodes for, 235–239 , 235–240	animation curves, 309–315 , <i>310</i>
UV mapping, 232	anticipation, 308–310 , <i>310</i>
Surfaces menu set, 41	follow-through, 312–315 , <i>313–315</i>
symmetry	keyframes for, 313–315
meshes, 49 , 49	path animation, 317–318, 317
toy airplane wings, 207–209, 207–208	playing back, 312–313
Symmetry selections for polygons, 107–108, 107–108	preproduction process, 307
syntax, 11	replacing axe, 318–319
	secondary motion, 315–317, 316
т	thumbs, 96–98 , <i>96–99</i>
	TIFF (Tagged Image File Format) files, 5, 436
T Divisions attribute, 167	Time settings, 558
tablecloth, 552–554 , <i>552–554</i>	Time Slider, 11, 26, 26, 41, 70–71 , 70
Tagged Image File Format (TIFF) files, 5, 436	Time Snap icon, 311
tangency	Timeline scrubbing, 27
constraints, 371	timing for bouncing ball, 304–305 , <i>304</i>
handles for, 300, 301, 314, 314	toe joints, 332, 333
tapers for catapult axles, 129, 130	Tool Box, 14, 14, 40, 67 , 67, 559
Targa format, 5, 436	Tool Settings window, 47, 48
tessellation, 152–153	toon shading, 291–293 , <i>292–293</i>
text, flying, 320–322 , <i>320–323</i>	torus for rings, 20
Texture Display mode, 243	toy airplane. See also living room scene
texture shaded mode, 45	assembling, 215-217, 216-217
texturing and textures, 219–220	body, 194–197 , <i>194–197</i>
3D, 239	engine, 201–204 , <i>202–204</i>
importing image files as, 237-238, 237	overview, 176 , <i>176–177</i>
mapping, 220	pontoons
nodes, 235–239 , <i>235–240</i>	arms, 188–194 , <i>189–193</i>
photo-real mapping, 274	in assembly, 215-216, 215-216
color, 282–290 , <i>282–291</i>	blocking out, 184–186 , <i>184–186</i>
feet, 274–282 , <i>275–281</i>	reference images, 178–184 , <i>178–184</i>
projections for, 232–235 , 233–234	shaping, 186–188 , <i>186–188</i>
red wagon, 240 , 240	propeller blades, 205–206 , <i>205–206</i>
finishing, 273–274 , 273–274	rear stabilizers, 198–201 , <i>198–201</i>
floor, 270 , <i>270</i>	scene organization and geometry, 211–212 , <i>211–212</i>
railings, 270–273 , <i>271–273</i>	shading and texturing, 227–231 , 227–231
shader assignment, 241–243 , <i>241–243</i>	
wheels, 243–252 , 243–252	summary, 217
summary, 294	wing supports, 212–214, 212–214
Julii y, 47 1	wings, 205–210 , 206–211

train animating, 538 – 547 , <i>539</i> – <i>546</i> pumps, 375, 379 , <i>379</i> rigging. <i>See</i> rigging Transfer Attribute Options dialog box, 263, 264 transferring animation, 319 , <i>319</i> transform nodes, 28–29 Transform tools, 13–14, <i>14</i> transformations, freezing, 296–297, 316–317 Translate Frame setting, 235 Translate keyframes axe throwing, 309–312, 315 bouncing ball, 299 translations and Translate attributes axe throwing, 315–316 baking out simulations, 526, 528 bouncing ball, 297–298, 301 IK walk cycle, 364 nParticles, 534 point constraints, 367	Up Axis setting, 321 Up Vector setting, 368–369 Use Background shader, 220 Use Default Material option, 243 Use Depth Map Shadows option, 407, 420 Use Image Sequence option, 455 Use Ray Trace Shadows option, 408 User Interface. See interface UV coordinates, 254, 254 copying, 263–264 decorative box color, 282–290, 282–291 feet, 274–282, 275–281 mapping, 232 Photoshop, 259–260, 259–261 red wagon panels, 254–269, 254–269 UV Snapshot window, 259, 259, 282–283, 282 UV Texture Editor decorative box photo-real mapping, 289–290, 289–291
set driven keys, 372 Translucence attribute, 226 Translucence Focus attribute, 226 Transparency attributes nParticles, 546	photo-real mapping, 289–290, 289–291 UV setup, 274–282, 275–281 red wagon, 255–259, 255–257, 264–265 UVSnapShot layer, 238
Paint Effects, 550 shaders, 224 triangles, 81 Triangulate option, 84 Trim option for text, 320 trucking, 451 true reflections, 439 Tube Shading attribute, 550 Tube Width1 and Width2 attribute, 551 Tubes per Step attribute, 551 Twist attribute, 385–386, 385 two-node cameras, 449, 450 2D blur process, 456 2D texture placement, 235–236, 235 24-bit color images, 4 Type Creation option, 320 Type menu for lighting, 399 Type setting ramp gradients, 236 text, 320	V Wave attribute, 236 vector constraints, 371 vector math, 3 vector rendering, 441-443, 442 Venus, 31-32 versions, saving, 21 vertices attaching to skeletons, 352 in NURBS, 144-146, 144 polygon, 81, 81 spheres, 47-49, 47-48 ViewCube, 12, 43 views keyboard shortcuts for, 43 normalizing and denormalizing, 310, 310 Visor window, 548, 548 volume, emitter, 532, 533 Volume Light Dir attribute, 404 Volume lights, 404-405, 404
U	volumetric lighting, 418–419 , 418
U Divisions attribute, 167 U Wave attribute, 236 UI. See interface underlighting, 394, 394 Universal Manipulator tool, 14, 46, 46	walk cycles Forward Kinematics, 341–348 , <i>341</i> , <i>343–345</i> , <i>347</i> Inverse Kinematics, 363–365 , <i>364–365</i> Water texture, 237